

Michael Kofler



# Linux Kommandoreferenz

Shell-Befehle von A bis Z



- ▶ Weit über 400 wichtige Linux-Befehle immer griffbereit
- ▶ Schnell thematisch oder alphabetisch nachschlagen
- ▶ Viele Praxisbeispiele und spezifische Raspberry-Pi-Kommandos

**2., aktualisierte und erweiterte Auflage**



**Rheinwerk**  
Computing

## Liebe Leserin, lieber Leser,

vermutlich haben Sie bereits selbst die Erfahrung gemacht – über die Shell lässt sich Linux mit ein wenig Übung äußerst schnell und effizient steuern. Die Möglichkeiten der Steuerung und Automatisierung verschiedener Abläufe mithilfe von Shell-Befehlen gehen dabei weit über die Funktionen hinaus, die Ihnen die verschiedenen Benutzeroberflächen grafisch zur Verfügung stellen. Kurz: Wenn Sie den Umgang mit Shell-Kommandos beherrschen, dann liegt Ihnen Linux zu Füßen.

Die Menge der Möglichkeiten hat natürlich eine Kehrseite: nämlich die Fülle der damit verbundenen Informationen. Es gibt sehr viele Linux-Kommandos und eine schier unüberschaubare Anzahl von Optionen – ein Blick in die `man`-Seiten zu den einzelnen Kommandos spricht Bände ...

Und das bringt uns zu dieser Kommandoreferenz. Sie begrenzt und ordnet gezielt die Menge der Informationen und ermöglicht Ihnen so den schnellen Zugriff auf die wichtigsten Kommandos und häufig benötigte Optionen. Darüber hinaus gibt sie Ihnen geprüfte und sofort einsetzbare Anwendungsbeispiele an die Hand, weist Sie auf distributionsspezifische Besonderheiten hin und bietet Ihnen neben einer Übersicht wichtiger Tastenkürzel auch eine Syntaxreferenz zentraler Linux-Konfigurationsdateien. Und in dieser Auflage hat Michael Kofler, dessen Linux-Standardwerk Sie vermutlich kennen, zahlreiche weitere Kommandos für Sie beschrieben. Speziell für Fans des Mini-Linux-Superhelden Raspberry Pi finden sich darüber hinaus in dieser Kommandoreferenz ab sofort spezifische Pi-Kommandos zur Konsolensteuerung. Sie sehen: Noch ein Grund mehr, dieses Buch immer in Reichweite Ihres (Mini-) Linux-Rechners zu platzieren!

Noch ein Wort in eigener Sache: Dieses Werk wurde mit großer Sorgfalt geschrieben, geprüft und produziert. Sollte dennoch einmal etwas nicht so funktionieren, wie Sie es erwarten, freue ich mich, wenn Sie sich mit mir in Verbindung setzen. Ihre Kritik und konstruktiven Anregungen sind uns jederzeit herzlich willkommen!

**Ihr Sebastian Kestel**

Lektorat Rheinwerk Computing

sebastian.kestel@rheinwerk-verlag.de

www.rheinwerk-verlag.de

Rheinwerk Verlag · Rheinwerkallee 4 · 53227 Bonn

## Hinweise zur Benutzung

Dieses E-Book ist **urheberrechtlich geschützt**. Mit dem Erwerb des E-Books haben Sie sich verpflichtet, die Urheberrechte anzuerkennen und einzuhalten. Sie sind berechtigt, dieses E-Book für persönliche Zwecke zu nutzen. Sie dürfen es auch ausdrucken und kopieren, aber auch dies nur für den persönlichen Gebrauch. Die Weitergabe einer elektronischen oder gedruckten Kopie an Dritte ist dagegen nicht erlaubt, weder ganz noch in Teilen. Und auch nicht eine Veröffentlichung im Internet oder in einem Firmennetzwerk.

Die ausführlichen und rechtlich verbindlichen Nutzungsbedingungen lesen Sie im Abschnitt Rechtliche Hinweise.

## Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

**Lektorat** Sebastian Kestel, Christoph Meister

**Korrektorat** Friederike Daenecke, Zülpich

**Herstellung E-Book** Norbert Englert

**Covergestaltung** Mai Loan Nguyen Duy

**Coverbilder** Mai Loan Nguyen Duy; iStockphoto: 13845349 © Heike Brauer

**Satz E-Book** Michael Kofler

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Bitte teilen Sie uns doch Ihre Meinung mit und lesen Sie weiter auf den [Serviceseiten](#).

Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

**ISBN 978-3-8362-3778-9 (Buch)**

**ISBN 978-3-8362-3779-6 (E-Book)**

© Rheinwerk Verlag GmbH, Bonn 2016

3., aktualisierte und erweiterte Auflage 2016 (2. Auflage im Rheinwerk Verlag)

# Thematische Kommandoübersicht

## Dateiverwaltung

cat	verbindet mehrere Dateien zu einer Gesamtdatei .....	52
cd	wechselt in ein anderes Verzeichnis .....	52
chgrp	ändert die Gruppenzugehörigkeit einer Datei .....	57
chmod	ändert die Zugriffsbits einer Datei .....	59
chown	ändert den Besitzer einer Datei .....	60
cp	kopiert Dateien .....	66
dircolors	hilft bei der Konfiguration der ls-Farben .....	82
file	versucht, den Typ einer Datei festzustellen .....	102
inotifywait	wartet auf die Veränderung von Dateiattributen .....	156
j	wechselt in ein anderes Verzeichnis .....	171
ln	stellt feste und symbolische Links zu Dateien her .....	183
ls	zeigt das Inhaltsverzeichnis an .....	189
mkdir	erzeugt ein neues Verzeichnis .....	206
mv	verschiebt Dateien bzw. ändert ihren Namen .....	234
namei	zeigt die Zugriffsrechte aller Verzeichnisse einer Datei .....	241
rename	gibt mehreren Dateien neue Namen .....	282
rm	löscht Dateien .....	286
rmdir	löscht Verzeichnisse .....	287
rsync	synchronisiert Verzeichnisse .....	292
stat	liefert detaillierte Informationen zu Dateien .....	323
tee	dupliziert die Standardeingabe .....	336
touch	erzeugt eine leere Datei bzw. ändert den Modify-Zeitpunkt .....	342
tree	stellt die Verzeichnishierarchie dar .....	345
truncate	verkleinert/vergrößert eine Datei .....	345
umask	steuert, welche Zugriffsrechte neue Dateien und Verzeichnisse erhalten .....	350

## Access Control Lists (ACLs) und Extended Attributes (EAs)

attr	verwaltet die Zusatzattribute einer Datei .....	42
chacl	verwaltet die ACLs einer Datei .....	53
getcap	ermittelt die Capabilities einer Datei .....	122

getfacl	ermittelt die ACLs einer Datei .....	123
getfattr	ermittelt die Zusatzattribute einer Datei .....	123
setcap	verändert die Capabilities einer Datei .....	299
setfacl	verändert die ACLs einer Datei .....	300
setfattr	verändert die Zusatzattribute einer Datei .....	303

## Dateien suchen

---

find	sucht Dateien nach Name, Datum, Größe etc. ....	102
grep	sucht Text in einer Textdatei .....	135
locate	sucht Dateien in einer dafür vorbereiteten Datenbank .....	185
updatedb	aktualisiert die Suchdatenbank für locate .....	356
whereis	sucht Dateien in vordefinierten Verzeichnissen .....	376
which	durchsucht die PATH-Verzeichnisse nach Kommandos .....	376

## Bearbeitung von Textdateien

---

awk	Programmiersprache zur Textauswertung .....	42
cat	gibt eine Datei aus bzw. vereint mehrere Texte .....	52
csplit	zerlegt den Text an vorgegebenen Stellen in Einzeldateien	71
cut	extrahiert Spalten aus jeder Zeile des Textes .....	74
diff	vergleicht zwei Texte .....	81
expand	ersetzt Tabulator- durch Leerzeichen .....	97
fold	zerlegt lange Textzeilen in kürzere .....	110
grep	sucht Texte innerhalb der Datei .....	135
head	gibt die ersten Zeilen der Datei aus .....	149
iconv	ändert den Zeichensatz von Textdateien .....	152
less	zeigt Textdateien seitenweise an (auch rückwärts) .....	181
more	zeigt Textdateien seitenweise an .....	215
multitail	verfolgt die Änderungen mehrerer Dateien .....	234
paste	vereint mehrere Texte zeilenweise .....	261
patch	ändert Textdateien gemäß einer diff-Datei .....	261
recode	konvertiert zwischen verschiedenen Zeichensätzen .....	281
sed	Stream-Editor (programmierbarer Editor) .....	296
sort	sortiert Dateien .....	316
split	zerlegt eine Datei in Teildateien mit vorgegebener Größe	318
strings	extrahiert Zeichenketten aus Binärdateien .....	324

<code>tac</code>	gibt Text in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst .....	331
<code>tail</code>	gibt das Ende einer Datei aus .....	331
<code>tr</code>	ersetzt vorgegebene Zeichen durch andere Zeichen .....	343
<code>unexpand</code>	ersetzt Leerzeichen durch Tabulatorzeichen .....	352
<code>uniq</code>	eliminiert mehrfach auftretende Zeilen in einer Textdatei .....	352
<code>zcat</code>	gibt eine komprimierte Textdatei aus .....	390
<code>zless</code>	zeigt eine komprimierte Textdatei an (auch rückwärts) .....	390
<code>zmore</code>	zeigt eine komprimierte Textdatei seitenweise an .....	390

## Dateien komprimieren und archivieren

<code>bunzip2</code>	dekomprimiert *.bz2-Dateien .....	50
<code>bzip2</code>	komprimiert Dateien; leistungsfähiger als <code>gzip</code> .....	50
<code>compress</code>	komprimiert Dateien .....	63
<code>cpio</code>	überträgt Archivdateien zwischen Dateisystemen .....	68
<code>gunzip</code>	dekomprimiert *.gz-Dateien .....	144
<code>gzip</code>	komprimiert Dateien; leistungsfähiger als <code>compress</code> .....	145
<code>lzop</code>	komprimiert Dateien; deutlich schneller als <code>gzip</code> .....	200
<code>mt</code>	steuert den Streamer (Vor- und Rückspulen etc.) .....	233
<code>tar</code>	bildet ein *.tar-Archiv .....	332
<code>uncompress</code>	dekomprimiert durch <code>compress</code> komprimierte Dateien .....	352
<code>unxz</code>	dekomprimiert *.xz-Dateien .....	353
<code>unzip</code>	dekomprimiert *.zip-Dateien .....	353
<code>xz</code>	komprimiert Dateien; leistungsfähiger als <code>bzip2</code> .....	386
<code>zip</code>	erzeugt ein Windows-kompatibles ZIP-Archiv .....	391
<code>zipinfo</code>	zeigt den Inhalt eines ZIP-Archivs an .....	392

## CDs/DVDs schreiben

<code>cdrecord</code>	schreibt eine CD im Disk-at-once-Modus .....	53
<code>cdrecord</code>	schreibt eine CD oder DVD .....	53
<code>dvd+rw-format</code>	formatiert DVD+RW- und DVD-RW-Medien .....	89
<code>dvd+rw-mediainfo</code>	liefert Informationen über ein optisches Datenmedium .....	89
<code>genisoimage</code>	erzeugt ein ISO-Image (ehemals <code>mkisofs</code> ) .....	118
<code>growisofs</code>	schreibt DVDs und Blu-ray-Discs .....	140
<code>isohybrid</code>	macht ein ISO-Image für USB-Datenträger bootfähig .....	168

mkisofs	erzeugt ein ISO-Image .....	118
mkudffs	richtet ein UDF-Dateisystem ein (DVD-RAM) .....	214
wodim	schreibt eine CD oder DVD (ehemals cdrecord) .....	378

## Konverter

---

a2ps	konvertiert Textdateien in das PostScript-Format .....	25
avconv	konvertiert Video-Dateien .....	42
convert	konvertiert Grafikdateien .....	63
convmv	ändert den Zeichensatz von Dateinamen .....	65
dvisp	wandelt eine DVI-Datei in das PostScript-Format um .....	89
enscript	konvertiert Textdateien in das PostScript-Format .....	94
epstopdf	konvertiert EPS-Dateien in PDF-Dateien .....	95
ffmpeg	konvertiert Video-Dateien .....	100
iconv	ändert den Zeichensatz von Textdateien .....	152
lame	erzeugt MP3-kompatible Audio-Dateien .....	180
mogrify	verändert Parameter einer Bilddatei .....	215
mpage	konvertiert Textdateien in das PostScript-Format .....	232
pandoc	erzeugt Dokumente aus Markdown-Dateien .....	252
paps	konvertiert UTF-8-Textdateien in das PostScript-Format .....	254
pdf2ps	konvertiert PDF-Dateien in PostScript-Dateien .....	262
pdftk	manipuliert PDF-Dateien .....	262
pdftops	Alternative zu pdf2ps .....	264
pdftotext	wandelt ein PDF-Dokument in eine reine Textdatei um .....	265
ps2pdf	konvertiert PostScript-Dateien in PDF-Dateien .....	269
recode	ändert den Zeichensatz von Textdateien .....	281

## Programm-/Prozessverwaltung

---

at	führt einen Job zu einem vordefinierten Zeitpunkt aus .....	39
atq	listet Jobs auf, die später ausgeführt werden sollen .....	42
atrm	löscht einen Job, der später ausgeführt werden soll .....	42
batch	führt einen Job aus, sobald das System im Leerlauf läuft .....	45
bg	setzt einen Prozess im Hintergrund fort .....	45
chroot	startet eine Shell in einem veränderten Wurzel- verzeichnis .....	61
crontab	hilft bei der Administration eigener Crontab-Einträge .....	69
disown	löst einen Prozess von der Shell .....	83



fg	setzt einen Prozess im Vordergrund fort .....	102
fuser	ermittelt das Programm, das auf eine Datei zugreift .....	115
halt	beendet Linux und hält den Rechner an .....	145
htop	zeigt alle fünf Sekunden eine Liste aller Prozesse an .....	151
iftop	beobachtet die Netzwerkaktivität .....	154
ionice	steuert die I/O-Priorität eines Prozesses .....	160
iotop	zeigt die Prozesse mit der größten IO-Aktivität an .....	160
kill	versendet Signale (meist zum Beenden von Prozessen) .....	173
killall	wie kill; der Prozess wird mit Namen genannt .....	173
ldconfig	aktualisiert die Cache-Datei zur Bibliothekssuche .....	181
ldd	liefert alle erforderlichen Bibliotheken eines Programms ....	181
lsof	listet offene Dateien und die zugeordneten Prozesse auf ....	195
nice	startet ein Programm mit verringerter Priorität .....	244
nohup	startet einen Prozess, der von der Shell losgelöst ist .....	247
pidof	ermittelt die Prozessnummer eines Programms .....	265
powertop	analysiert den Energieverbrauch der laufenden Prozesse ....	266
ps	zeigt die Liste der laufenden Prozesse an .....	268
pstree	wie ps; macht die Abhängigkeiten besser sichtbar .....	271
reboot	beendet Linux und startet den Rechner neu .....	281
renice	verändert die Priorität eines laufenden Prozesses .....	283
sudo	führt ein Programm als root aus .....	325
timeout	limitiert die Ausführungszeit eines Kommandos .....	341
top	zeigt alle fünf Sekunden eine Liste aller Prozesse an .....	342
watch	führt ein Kommando periodisch aus und zeigt die Ausgaben an .....	372

## Verwaltung von Benutzern und Gruppen

addgroup	richtet eine neue Gruppe ein (Debian/Ubuntu) .....	28
adduser	richtet einen neuen Benutzer ein (Debian/Ubuntu) .....	28
chage	ändert das Ablaufdatum eines Kontos oder Passworts .....	54
chgrp	ändert die Gruppenzugehörigkeit einer Datei .....	57
chown	ändert den Besitzer einer Datei .....	60
chsh	verändert die Default-Shell eines Benutzers .....	61
delgroup	löscht eine Gruppe (Debian/Ubuntu) .....	79
deluser	löscht einen Benutzer (Debian/Ubuntu) .....	79
groupadd	richtet eine neue Gruppe ein .....	139
groupdel	löscht eine Gruppe .....	139
groupmod	verändert Gruppeneigenschaften .....	140
groups	zeigt die Gruppen des aktuellen Benutzers an .....	140

id	zeigt die aktuelle Benutzer- und Gruppen-ID-Nummer an	153
last	verrät, wer zuletzt auf diesem Rechner eingeloggt war	180
lastb	listet auf, welche Login-Versuche zuletzt gescheitert sind	181
makepasswd	erzeugt ein neues, zufälliges Passwort	202
mkpasswd	erzeugt ein neues, zufälliges Passwort	212
newgrp	ändert die aktive Gruppe eines Benutzers	243
newusers	richtet mehrere neue Benutzer ein	243
passwd	verändert das Passwort eines Benutzers	260
pwgen	generiert leicht merkbare Passwörter	272
useradd	richtet einen neuen Benutzer ein	356
userdel	löscht einen Benutzer	357
usermod	verändert Benutzereigenschaften	357
who	liefert Informationen über die eingeloggten Benutzer	377

## Administration des Dateisystems

---

badblocks	testet, ob Datenträger defekte Sektoren enthalten	44
blkid	liefert die UUID und den Namen eines Dateisystems	45
btrfs	administriert ein btrfs-Dateisystem	47
cdisk	partitioniert eine Festplatte	53
cryptsetup	richtet ein Crypto-Device ein	70
dcfldd	kopiert Datenblöcke zwischen Devices (mit Fortschritts- anzeige)	76
dd	kopiert Datenblöcke zwischen Devices	77
df	zeigt den freien Speicher auf der Festplatte an	80
du	ermittelt den Platzbedarf eines Verzeichnisses	87
dumpe2fs	zeigt interne Informationen über ein ext-Dateisystem an	88
e4defrag	defragmentiert Dateien eines ext4-Dateisystems	90
eject	wirft eine CD oder Diskette aus	93
exfatlabel	ändert den Namen eines exFAT-Dateisystems	96
fdisk	partitioniert eine Festplatte mit MBR-Partitionstabelle	98
findmnt	liefert eine Liste aller aktiven Dateisysteme	106
fstrim	meldet der SSD alle freien Datenblöcke	113
gdisk	partitioniert eine Festplatte mit GUID-Partitionstabelle	117
hdparm	verändert Parameter der Festplatte	148
kpartx	erzeugt oder löscht Device-Dateien für virtuelle Datenträger	174
lsblk	listet alle Block-Devices auf	193
luksformat	richtet ein Crypto-Device samt Dateisystem ein	197
mdadm	verwaltet RAID-Partitionen	203

mkfifo	erzeugt eine FIFO-Datei (eine benannte Pipe) .....	206
mkfs	richtet ein Dateisystem ein .....	207
mknod	erstellt Device-Dateien .....	212
mkswap	richtet eine Datei oder eine Partition als Swap-Bereich ein .....	213
mount	bindet ein Dateisystem in den Verzeichnisbaum ein .....	215
parted	partitioniert eine Festplatte .....	255
partprobe	informiert den Kernel über die geänderte Partitionierung .....	258
partx	liest Partitionen bzw. verändert die Partitionstabelle des Kernels .....	258
resize2fs	verändert die Größe eines ext-Dateisystems .....	284
sfdisk	partitioniert eine Festplatte mit MBR-Partitionstabelle .....	306
sgdisk	partitioniert eine Festplatte mit GUID-Partitionstabelle .....	307
smartctl	steuert die SMART-Funktionen der Festplatte .....	311
snapper	verwaltet btrfs-Snapshots (SUSE) .....	315
swapoff	deaktiviert eine Swap-Datei oder -Partition .....	328
swapon	aktiviert eine Swap-Datei oder -Partition .....	328
sync	führt alle gepufferten Schreiboperationen aus .....	329
tune2fs	verändert Systemparameter eines ext-Dateisystems .....	346
umount	löst ein Dateisystem aus dem Verzeichnisbaum .....	351
volname	liefert den Partitionsnamen einer CD .....	372
xfs_admin	verändert Parameter eines XFS-Dateisystems .....	382
xfs_growfs	vergrößert ein XFS-Dateisystem .....	382
xfs_info	zeigt die Eckdaten eines XFS-Dateisystems .....	383
xfs_repair	repariert ein defektes XFS-Dateisystem .....	383

## LVM-Kommandos

lvcreate	richtet ein neues LV (Logical Volume) ein .....	197
lvdisplay	liefert Detailinformationen zu einem LV .....	198
lvextend	vergrößert ein LV .....	198
lvm	LVM-Basiskommando .....	199
lvreduce	verkleinert ein LV .....	199
lvremove	löscht ein LV .....	199
lvrename	gibt dem LV einen neuen Namen .....	200
lvscan	listet alle LVs auf .....	200
pvcreate	kennzeichnet eine Partition als PV (Physical Volume) .....	272
pvdisk	liefert Detailinformationen zu einem PV .....	272
pvremove	entfernt die PV-Kennzeichnung eines ungenutzten PVs .....	272
pvscan	listet alle PVs auf .....	272
vgchange	ändert die Attribute einer VG (Volume Group) .....	359

vgcreate	erzeugt eine neue VG aus einem oder mehreren PVs .....	359
vgdisplay	liefert Detailinformationen zu einer VG .....	359
vgextend	vergrößert eine VG um ein PV .....	359
vgmerge	vereint zwei VGs .....	360
vgreduce	verkleinert eine VG um ein ungenutztes PV .....	360
vgrename	gibt einer VG einen neuen Namen .....	360
vgscan	listet alle VGs auf .....	360

## SELinux und AppArmor

---

aa-complain	protokolliert AppArmor-Regelverstöße, ohne sie zu ahnden .....	26
aa-disable	deaktiviert ein AppArmor-Regelprofil .....	26
aa-enforce	stellt die Einhaltung von AppArmor-Regeln sicher .....	26
aa-status	ermittelt den Zustand des AppArmor-Systems .....	26
chcon	verändert den SELinux-Kontext von Dateien .....	56
getenforce	ermittelt den SELinux-Modus (Enforcing oder Permissive) .....	122
restorecon	stellt den Default-SELinux-Kontext wieder her .....	285
sealert	hilft bei der Analyse von SELinux-Regelverstößen .....	296
sestatus	ermittelt den Zustand des SELinux-Systems .....	298
setenforce	ändert den SELinux-Modus zwischen Enforcing und Permissive .....	300
setsebool	verändert boolesche Parameter der SELinux-Regeln .....	304

## Paketverwaltung

---

add-apt-repository	richtet eine PPA-Paketquelle ein (Ubuntu) .....	27
alien	wandelt Pakete zwischen verschiedenen Formaten um .....	29
alternatives	richtet Links in /etc/alternatives ein (Fedora, Red Hat) .....	31
apt-cache	liefert Informationen über installierte/verfügbare Pakete .....	34
apt-cdrom	richtet eine CD/DVD als APT-Paketquelle ein .....	35
apt-get	hilft bei der DEB-Paketverwaltung (Debian, Ubuntu) .....	35
apt-key	richtet einen Schlüssel für eine APT-Paketquelle ein .....	37
aptitude	hilft bei der DEB-Paketverwaltung .....	38
cnf	verrät, welches Paket ein Kommando enthält (SUSE) .....	63
dnf	Alternative zu yum (Fedora) .....	84
dpkg	(de)installiert oder aktualisiert DEB-Pakete .....	84
ppa-purge	deaktiviert eine PPA-Paketquelle (Ubuntu) .....	267

repoquery	liefert Metadaten aus YUM-Paketquellen .....	283
rpm	(de)installiert oder aktualisiert RPM-Pakete .....	289
rpm2archive	wandelt ein Paket in ein TAR-Archiv um .....	292
rpm2cpio	wandelt ein Paket in ein CPIO-Archiv um .....	292
taskel	(de)installiert DEB-Paketgruppen .....	336
update-alternatives	richtet Links in /etc/alternatives ein .....	354
yum	hilft bei der RPM-Paketverwaltung (Fedora, Red Hat) .....	387
yumdownloader	lädt den Quellcode von Paketen herunter .....	390
zypper	hilft bei der RPM-Paketverwaltung (SUSE) .....	392

## Netzwerkadministration

curl	überträgt Dateien von/zu HTTP-, FTP- und SSH-Servern .....	73
dhclient	führt eine DHCP-Netzwerkconfiguration durch .....	81
ethtool	verändert die Parameter eines Ethernet-Adapters .....	95
exportfs	meldet die NFS-Konfiguration an den NFS-Server .....	97
firewall-cmd	liest bzw. ändert die Firewall-Konfiguration (RHEL/Fedora) .....	108
ftp	überträgt interaktiv Dateien via FTP .....	114
host	löst IP-Nummern bzw. Netzwerknamen auf .....	149
hostname	liefert bzw. verändert den Namen des lokalen Rechners .....	150
hostnamectl	verändert den Hostnamen bleibend .....	151
ifconfig	konfiguriert Netzwerkschnittstellen .....	154
ifdown	deaktiviert eine Netzwerkschnittstelle .....	155
ifup	aktiviert eine Netzwerkschnittstelle .....	155
ip	zeigt Netzwerkeinstellungen an bzw. verändert sie .....	161
ipcalc	berechnet Netzwerkbereiche und -masken .....	163
iptables	konfiguriert den Linux-Paketfilter (Firewall) .....	164
iptables-restore	liest mehrere Firewall-Regeln ein .....	167
iptables-save	gibt alle Firewall-Regeln im Textformat aus .....	167
iptables-xml	gibt alle Firewall-Regeln als XML-Dokument aus .....	167
iw	steuert WLAN-Schnittstellen (aktuelles Kommando) .....	168
iwconfig	steuert WLAN-Schnittstellen (veraltetes Kommando) .....	170
iwlist	liefert Informationen zum WLAN-Controller und -Netz .....	170
mtr	kombiniert ping- und traceroute-Ergebnisse .....	233
netstat	analysiert die Netzwerkaktivität des lokalen Rechners .....	241
newaliases	meldet Änderungen in /etc/aliases an den Mail-Server .....	242
nmap	analysiert die Netzwerkaktivität eines fremden Rechners ...	245
nmcli	steuert den Network Manager .....	245
openssl	erzeugt und administriert SSL-Schlüsseldateien .....	248

ping	überprüft die Netzwerkverbindung zu einem Rechner .....	266
pnuke	beendet ein Programm parallel auf mehreren Hosts .....	270
pscp	kopiert Dateien parallel von/auf mehrere(n) Hosts .....	270
pssh	führt via SSH Kommandos parallel auf mehreren Hosts aus .....	270
rdiff-backup	erstellt inkrementelle Backups .....	279
rftkill	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter .....	285
route	verändert bzw. zeigt die IP-Routing-Tabelle .....	288
rpcinfo	liefert Informationen über RPC- und NFS-Dienste .....	289
rsync	synchronisiert Netzwerkverzeichnisse .....	292
scp	überträgt Dateien verschlüsselt via SSH .....	294
sftp	überträgt Dateien via SFTP .....	307
showmount	listet NFS-Verzeichnisse auf .....	309
smbclient	überträgt Dateien aus Windows-Netzwerkverzeichnissen .....	312
smbtree	liefert eine Liste aller Windows-Netzwerkverzeichnisse .....	313
ss	analysiert die Netzwerkaktivität des lokalen Rechners .....	319
ssh	ermöglicht Logins auf anderen Rechnern im Netzwerk .....	320
ssh-copy-id	überträgt einen öffentlichen Schlüssel zum SSH-Server .....	322
telnet	kommuniziert interaktiv mit einem Netzwerkdienst .....	337
traceroute	liefert die Zwischenstationen zu einer Netzwerkadresse .....	343
ufw	konfiguriert die Firewall (Ubuntu) .....	348
wget	lädt Dateien oder Verzeichnisse herunter .....	373

## Drucker-, Datenbank- und Server-Administration

---

htpasswd	speichert Apache-Login-Daten in einer Passwortdatei .....	152
lpadmin	richtet neue Drucker ein bzw. löscht sie wieder .....	186
lpinfo	listet alle Druck-Devices, Druckertreiber etc. auf .....	187
lpoptions	zeigt die Optionen von Druckern an bzw. verändert sie .....	187
lpq	zeigt den Inhalt einer Druckerwarteschlange an .....	188
lpr	druckt eine Datei aus .....	188
lprm	löscht einen Druck-Job aus der Warteschlange .....	188
lpstat	liefert Informationen über Drucker, Druck-Jobs etc. .....	189
mysql	führt SQL-Kommandos auf einem MySQL-Server aus .....	235
mysqladmin	hilft bei der MySQL-Administration .....	237
mysqldump	führt ein MySQL-Backup durch .....	239
smbpasswd	ändert ein Samba-Passwort .....	313
sqlite3	führt SQL-Kommandos in SQLite-Datenbanken aus .....	319

## Audio-Funktionen und Hardware-Verwaltung

acpi	liefert Informationen über den Batteriezustand .....	27
alsactl	speichert bzw. lädt alle Parameter des Audio-Systems .....	30
alsamixer	stellt die Audio-Kanäle interaktiv ein .....	31
amixer	steuert die Audio-Kanäle durch Optionen .....	31
free	zeigt den freien Speicherplatz an (RAM/Swap) .....	111
localectl	verändert die Sprach- und Tastatureinstellungen .....	184
lscpu	liefert Informationen über die CPU .....	193
lshw	bildet eine hierarchische Liste der Hardware-Komponenten .....	194
lspci	liefert Informationen über PCI-Komponenten .....	196
lsscsi	liefert Informationen über angeschlossene SCSI-Geräte .....	196
lsusb	liefert Informationen über angeschlossene USB-Geräte .....	197
pactl	steuert den PulseAudio-Server .....	251
paplay	spielt eine RAW-Datei via PulseAudio .....	253
parecord	führt eine RAW-Audio-Aufnahme via PulseAudio durch .....	253
powertop	hilft den Energieverbrauch von Notebooks zu optimieren ...	266
speaker-test	testet das Audio-System und die angeschlossenen Lautsprecher .....	317
timedatectl	stellt Datum, Uhrzeit und Zeitzone ein .....	341
vcgencmd	liest bzw. verändert Hardware-Parameter des Raspberry Pi (Raspbian) .....	357

## Bluetooth

hciconfig	(de)aktiviert lokale Bluetooth-Adapter .....	146
hcitool	verwaltet Bluetooth-Geräte .....	147
l2ping	sendet Echo-Anfragen an Bluetooth-Geräte .....	179
rftkill	(de)aktiviert Bluetooth-, WLAN- und Mobilfunk-Adapter .....	285
sdptool	ermittelt Detailinformationen über Bluetooth-Geräte .....	295

## Systemstart und -stopp, Init-System, Logging und GRUB

chkconfig	richtet Init-V-Links ein (Red Hat, Fedora) .....	57
efibootmgr	liest bzw. verändert die Tabelle der EFI-Booteinträge .....	91
grub	startet die GRUB-Shell (GRUB 0.97 <i>legacy</i> ) .....	141
grub-install	installiert GRUB in den Bootsektor (GRUB 2) .....	142

grub-mkconfig	erzeugt eine neue GRUB-Konfigurationsdatei (GRUB 2) .....	143
init	wechselt in einen anderen Runlevel (Init-V-System) .....	156
initctl	steuert Upstart und generiert Upstart-Ereignisse .....	156
insserv	richtet Init-V-Links ein (Debian, SUSE) .....	158
invoke.rc	führt ein Init-V-Script aus (Debian) .....	159
journalctl	liest Nachrichten aus dem Journal .....	171
logger	protokolliert eine Nachricht .....	186
mkinitrd	erzeugt eine Initrd-Datei (Fedora, Red Hat, SUSE) .....	212
service	führt ein Init-V-Script aus .....	298
shutdown	beendet Linux .....	310
start	startet einen durch Upstart verwalteten Dienst .....	323
stop	stoppt einen durch Upstart verwalteten Dienst .....	323
systemctl	verwaltet Systemd-Prozesse .....	329
telinit	ändert den Runlevel .....	337
update-grub	aktualisiert die GRUB-Datei menu.lst (Debian) .....	354
update-initramfs	erzeugt eine Initrd-Datei (Debian, Ubuntu) .....	355

## Virtualisierung (libvirt, KVM)

---

kvm	führt eine virtuelle Maschine aus .....	175
qemu-img	erzeugt bzw. bearbeitet Image-Dateien .....	274
qemu-kvm	führt eine virtuelle Maschine aus .....	175
qemu-nbd	bietet eine Image-Datei als Netzwerk-Block-Device an .....	275
virsh	führt libvirt-Kommandos aus .....	360
virt-clone	erstellt eine Kopie einer virtuellen Maschine .....	367
virt-install	richtet eine neue virtuelle Maschine ein .....	368
virt-viewer	erlaubt die Bedienung einer virtuellen Maschine via VNC ...	371

## Terminal und Textkonsole

---

echo	gibt eine Zeile Text aus .....	91
loadkeys	lädt eine Tastaturliste für Textkonsolen .....	183
printf	ermöglicht eine formatierte Ausgabe wie unter C .....	268
reset	führt einen Reset für das Terminal durch .....	284
setfont	verändert die Terminal-Schriftart .....	304
setterm	verändert diverse Terminaleinstellungen .....	304



## Online-Hilfe

---

apropos	sucht Kommandos zu einem Thema .....	33
help	zeigt die Beschreibung eines Shell-Kommandos an .....	149
info	startet das info-System .....	155
man	zeigt die Beschreibung eines Kommandos an .....	202
whatis	zeigt eine Kurzbeschreibung eines Kommandos an .....	376

## Grafiksystem und Gnome

---

aticonfig	hilft bei der Konfiguration des AMD-Grafiktreibers .....	41
chvt	wechselt die aktive Textkonsole bzw. aktiviert das Grafiksystem .....	62
fc-list	listet alle skalierbaren Schriften auf .....	98
fgconsole	liefert die Nummer der aktiven Konsole .....	102
gconftool-2	liest bzw. ändert Einstellungen der gconf-Datenbank .....	117
glxinfo	überprüft die 3D-Funktionen des Grafiksystems .....	129
gnome-session-quit	initiiert einen Logout bzw. Shutdown .....	130
gsettings	liest bzw. ändert Einstellungen der dconf-Datenbank .....	143
gtf	berechnet die Parameter für einen neuen Grafikmodus .....	144
nvidia-xconfig	hilft bei der Konfiguration des NVIDIA-Grafiktreibers .....	248
tvservice	steuert den HDMI-Ausgang (Raspbian) .....	347
xdpinfo	liefert Informationen zum laufenden X-Server .....	381
xhost	erlaubt oder sperrt Hosts für den X-Login .....	383
xkill	beendet ein Programm per Mausclick .....	384
xrandr	ändert die Auflösung des Grafiksystems .....	384
xset	ändert Benutzereinstellungen des Grafiksystems .....	386
zenity	zeigt einfache grafische Dialoge an .....	391

## Sonstiges

---

alias	definiert eine Abkürzung .....	29
basename	ermittelt den Dateinamen eines Pfads .....	44
cksum	berechnet die CRC-Prüfsumme zu einer Datei .....	62
date	zeigt Datum und Uhrzeit an .....	75
dirname	ermittelt das Verzeichnis eines Pfads .....	83
dmesg	zeigt die Kernelmeldungen des Bootvorgangs an .....	84

expr	führt Berechnungen und Mustervergleiche durch .....	98
git	steuert das Versionsverwaltungssystem Git .....	124
gpio	steuert die Input/Output-Pins des Raspberry Pi (Raspbian) .....	134
hash	zeigt die Hash-Tabelle an .....	146
ldd	zeigt die erforderlichen Libraries für ein Programm an .....	181
mail	übergibt eine E-Mail an den lokalen Mail-Server .....	201
md5sum	berechnet eine Prüfsumme zu einer Datei .....	203
printenv	zeigt nur die Umgebungsvariablen an .....	268
qalc	ist ein Taschenrechner für das Terminal .....	273
raspi-gpio	steuert die Input/Output-Pins des Raspberry Pi (Raspbian) .....	276
raspistill	nimmt ein Foto auf (Raspbian) .....	277
raspivid	nimmt ein Video auf (Raspbian) .....	279
seq	liefert eine Zahlensequenz .....	297
set	zeigt alle der Shell bekannten Variablen an .....	299
sha512sum	berechnet eine Prüfsumme zu einer Datei .....	308
sleep	wartet eine vorgegebene Zeit .....	311
strace	verrät, welche Funktionen ein Programm aufruft .....	324
svn	steuert das Versionsverwaltungssystem Subversion .....	326
time	misst die Ausführzeit eines Kommandos .....	340
tty	zeigt den Device-Namen des Terminals an .....	346
type	gibt den Typ eines Kommandos an .....	347
unalias	löscht eine Abkürzung .....	351
uname	liefert den Betriebssystemnamen und die Kernelversion .....	351
xargs	leitet die Standardeingabe an ein Kommando weiter .....	380

## bash-Programmierung

---

break	beendet eine Schleife vorzeitig .....	46
case	leitet eine Fallunterscheidung ein .....	51
continue	überspringt den Schleifenkörper .....	63
exit	beendet das Shell-Programm .....	96
for	leitet eine Schleife ein .....	111
function	definiert eine neue Funktion .....	115
if	leitet eine Verzweigung ein .....	153
local	definiert lokale Variablen in einer Funktion .....	184
source	führt die angegebene Shell-Datei aus .....	317
test	wertet eine Bedingung aus .....	338
until	leitet eine Schleife ein (Variante 1) .....	353
while	leitet eine Schleife ein (Variante 2) .....	377

## bash-Variablenverwaltung

alias	definiert eine Abkürzung .....	29
declare	definiert eine (Umgebungs-)Variable .....	78
export	definiert eine Umgebungsvariable .....	97
local	definiert lokale Variablen in einer Funktion .....	184
read	liest eine Variable ein .....	281
readonly	zeigt alle schreibgeschützten Variablen an .....	281
shift	verschiebt die Parameterliste .....	309
unalias	löscht eine Abkürzung .....	351
unset	löscht eine Variable .....	353

## Weitere bash-Kommandos und -Sonderzeichen

dirs	zeigt die Liste der gespeicherten Verzeichnisse an .....	83
disown	löst einen Prozess von der Shell .....	83
eval	wertet das angegebene Kommando aus .....	96
popd	wechselt in das letzte gespeicherte Verzeichnis .....	266
pushd	speichert das Verzeichnis und wechselt in ein anderes .....	271
trap	führt beim Eintreten eines Signals ein Kommando aus .....	344
ulimit	kontrolliert die von der Shell beanspruchten Ressourcen .....	349
wait	wartet auf das Ende eines Hintergrundprozesses .....	372
---	bash-Sonderzeichen .....	395

## Konfigurationsdateien

adduser.conf	Einstellungen für neue Accounts (Debian, Ubuntu) .....	397
aliases	E-Mail-Weiterleitungen .....	399
bashrc	Defaulteinstellungen für die bash .....	400
config.txt	Raspberry-Pi-Hardware-Parameter (Raspbian) .....	400
crontab	Prozesse periodisch ausführen .....	402
deluser.conf	Einstellungen für deluser und delgroup (Debian, Ubuntu) .....	405
dnf.conf	Konfiguration der Paketverwaltung dnf (Fedora) .....	406
fstab	Dateisysteme/Partitionen automatisch einbinden .....	406
group	Gruppennamen und Gruppenzuordnungen .....	407
grub	Defaulteinstellungen für GRUB 2 .....	408
grub.cfg	Konfiguration für GRUB 2 .....	410

gshadow	Hash-Codes der Gruppenpasswörter .....	411
host.conf	Konfiguration der Resolver-Bibliothek .....	411
hostname	Hostname des Rechners .....	412
hosts	statische Liste von Hostnamen und IP-Adressen .....	412
ifcfg-xxx	Netzwerkparameter einer Schnittstelle (Fedora, Red Hat) ...	413
inittab	Defaulteinstellungen für das Init-V-System .....	416
interfaces	Netzwerkkonfiguration (Debian, Ubuntu) .....	418
journald.conf	Konfiguration des Logging-Diensts Journal .....	422
locale.conf	Lokalisierungseinstellungen (Systemd) .....	423
login.defs	Optionen für das Anlegen neuer Benutzer und Gruppen ....	424
mdadm.conf	Software-RAID-Konfiguration .....	426
menu.lst	Konfiguration für GRUB 0.97 .....	427
modules	Kernelmodule automatisch laden (Debian, Ubuntu) .....	428
network	Netzwerkgrundeinstellungen (Red Hat) .....	429
nsswitch.conf	Konfiguration der Name-Service-Switch-Funktionen .....	430
os-release	Namen und Versionsnummer der Distribution (Systemd) ...	431
passwd	Liste aller Konten, Login-Namen und Heimatverzeichnisse	431
profile	Konfiguration systemweiter Umgebungsvariablen .....	433
resolv.conf	IP-Adresse des Nameservers .....	433
rsyslog.conf	Konfiguration des Syslog-Dienstes .....	434
services	Zuordnung zwischen Netzwerkdiensten und Ports .....	437
shadow	Hash-Codes der Login-Passwörter .....	437
sources.list	APT-Paketquellen (Debian, Ubuntu) .....	439
sudoers	Konfiguration für sudo .....	439
sysctl.conf	Defaulteinstellungen für Kernelparameter .....	442
systemd.service	Konfiguration von Systemd-Diensten .....	442
systemd.timer	Konfiguration von periodischen Systemd-Jobs .....	446
vconsole.conf	Tastatureinstellungen (Systemd) .....	447
xorg.conf	Konfiguration des Grafiksystems .....	448
yum.conf	Konfiguration von yum (Fedora, Red Hat) .....	449

## Tastenkürzel

---

bash	Shell .....	453
emacs	Editor .....	453
fdisk	Partitionseditor .....	456
gnome-terminal	Terminal-Fenster unter Gnome .....	456
grub	Bootloader .....	457
info	Kommando zur Anzeige von Hilfetexten .....	458

joe	einfacher Editor .....	459
konsole	Terminal-Fenster unter KDE .....	459
less	Kommando zur Anzeige von Textdateien .....	460
man	Kommando zur Anzeige von Hilfetexten .....	461
mutt	E-Mail-Client für den Textmodus .....	461
nano	minimalistischer Editor .....	461
---	Tastenkürzel in Textkonsolen .....	462
vi	Editor .....	463



# Einleitung

Dieses Buch enthält Kurzbeschreibungen der wichtigsten Linux-Kommandos zur Verwaltung des Dateisystems, zum Starten und Beenden von Prozessen, zur Bearbeitung von Textdateien, für andere administrative Aufgaben sowie zur bash-Programmierung. Das Buch fasst außerdem die Syntax elementarer Konfigurationsdateien zusammen und enthält eine Tastenkürzelreferenz der Editoren Emacs, Nano und Vi sowie einiger anderer interaktiver Kommandos wie `less` oder `info`.

Das Ziel dieses Buches ist es, ein kompaktes Nachschlagewerk für die Arbeit mit Linux im Terminal zu bieten. Obwohl die Anzahl der Seiten in der hier vorliegenden Auflage weiter gestiegen ist, bleibt der Grundsatz dieses Buchs weiterhin: *Weniger ist mehr*. Dieses Buch kann und soll die `man`- und `info`-Seiten komplexer Kommandos nicht ersetzen! Exotische Optionen müssen Sie also weiterhin selbst nachsehen bzw. recherchieren.

Mit diesem Buch versuche ich aber, Ihnen die Arbeit abzunehmen, die oft über Dutzende Seiten reichende Originaldokumentation nach Optionen für den alltäglichen Gebrauch zu durchsuchen. Zahlreiche Beispiele zeigen zudem auf einen Blick den prinzipiellen Einsatz eines Kommandos.

Mitunter ist es so, dass man für eine bestimmte Aufgabe ein Kommando sucht, dessen Namen aber nicht kennt oder gerade vergessen hat. Für diese Fälle ist das thematisch organisierte Inhaltsverzeichnis gedacht.

Je nachdem, welche Distribution Sie einsetzen, stehen einige Kommandos standardmäßig nicht zur Verfügung und müssen extra installiert werden. Zudem gibt es distributionsspezifische Kommandos, die nur unter bestimmten Distributionen zur Verfügung stehen – z. B. die Paketverwaltungskommandos `dpkg` und `apt-get` (Debian, Ubuntu), `rpm` und `dnf` oder `yum` (Fedora, Red Hat) sowie `zypper` (SUSE). In der Kommandobeschreibung weise ich auf diesen Umstand jeweils hin.

## Was ist ein Kommando?

Linux unterscheidet nicht zwischen Kommandos, wie sie in diesem Buch beschrieben werden, und Programmen wie Firefox, LibreOffice oder Gimp. »Kommando« meint hier Programme ohne grafische Benutzeroberfläche, die in der Regel in einem Terminalfenster ausgeführt werden.

Ich beschreibe in diesem Buch auch einige Kommandos, die gar keine echten Programme sind, sondern nur Befehle der gerade aktiven Shell. Dabei gehe ich davon aus, dass Sie die `bash` (Bourne Again Shell) verwenden, die standardmäßig bei nahezu allen Linux-Distributionen für das interaktive Ausführen von Kommandos eingesetzt wird. Ein typisches Beispiel ist das häufig eingesetzte Kommando `cd` zum Wechseln des aktuellen Verzeichnisses.

### Optionen

Die meisten in diesem Buch beschriebenen Kommandos werden durch Optionen gesteuert. Die Angabe der Optionen erfolgt *vor* allen weiteren Parametern. Bei vielen Kommandos gibt es zwei Schreibweisen: `-x` für kurze Optionen (ein Buchstabe) und `--xyz` für lange Optionen (mehrere Buchstaben).

Die beiden folgenden `ls`-Kommandos sind gleichwertig und zeigen jeweils alle Dateien und Verzeichnisse im `/usr`-Verzeichnis an:

```
user$ ls -l -A /usr
user$ ls --format=long --almost-all /usr
```

Bei manchen Kommandos können mehrere Optionen als Gruppe angegeben werden (also `-ab` statt `-a -b`). Manche Kommandos kommen auch mit Optionen zurecht, die hinter dem oder den eigentlichen Parametern angegeben werden. Das sollte Sie aber nicht zu dem Schluss verleiten, dass das für alle Kommandos gilt!

```
user$ ls -lA /usr
user$ ls /usr -lA
```

Bei einigen wenigen Kommandos hat die Reihenfolge der Parameter einen Einfluss darauf, wie das Kommando ausgeführt wird. Wenn Optionen angegeben werden, die einander gegenseitig logisch ausschließen, gilt die zuletzt angegebene Option.

### man, info und help

Um dieses Buch nicht unnötig aufzublähen, beschreibe ich nur die wichtigsten Optionen. Eine vollständige Übersicht aller Optionen liefert bei der Mehrzahl der Kommandos `kommandoname --help`. Ausführlichere Informationen sind zumeist in den Manual-Seiten enthalten, die Sie mit `man name` bzw. mit `man 1 name` lesen können. Bei manchen Kommandos enthalten die `man`-Seiten lediglich einen Verweis auf die `info`-Texte, die entsprechend mit `info name` angezeigt werden.

Bei Kommandos, die direkt in die `bash` integriert sind (z. B. `cd`), führt `man name` zur `man`-Seite der `bash`. Dort ist das Kommando zwar tatsächlich beschrieben, aber die Suche in der sehr langen Dokumentation ist mühsam. Hilfreicher ist hier `help name`.



## Kommandos von A bis Z

```
a2ps [optionen] textdatei -o psdatei
```

a2ps wandelt Textdateien in das PostScript-Format um. Standardmäßig wird der Text über zwei Spalten auf eine Seite im Querformat verteilt. Bei Programmcode führt a2ps automatisch eine Syntaxhervorhebung durch.

- ▶ `-fn`  
stellt die Schriftgröße auf *n* Punkt ein. Damit ändert sich automatisch auch der Zeilenabstand und damit die Anzahl der Zeilen pro Seite. *n* ist eine Fließkommazahl.
- ▶ `--highlight-level=none`  
keine Syntaxhervorhebung durchführen.
- ▶ `--list=defaults`  
liefert eine Liste aller Standardeinstellungen von a2ps.
- ▶ `-M papierformat` bzw. `--medium=papierformat`  
verwendet das angegebene Papierformat (z. B. A3, A4, A5 oder Letter).
- ▶ `-o dateiname`  
speichert den resultierenden PostScript-Code in der angegebenen Datei. Ohne diese Option wird der Text direkt auf dem Standarddrucker ausgedruckt.
- ▶ `-R`  
Längsformat (Portrait) statt Querformat (Landscape).
- ▶ `-X charset`  
gibt den Zeichensatz des Texts an (z. B. ASCII, *ison*, *latin*).

a2ps kommt leider nicht mit Unicode-Textdateien zurecht. UTF-8-Textdateien müssen daher zuerst mit iconv oder recode in eine ISO-Codierung umgewandelt werden, bevor sie mit a2ps weiterverarbeitet werden können. Praktischer ist in solchen Fällen das Kommando paps, das von Haus aus UTF-8-kompatibel ist.

### Beispiel

Das folgende Kommando wandelt die angegebene Textdatei in eine PostScript-Datei um:

```
user$ a2ps datei.txt -o druck.ps
```

### **aa-complain** programm

Das Kommando `aa-complain` aktiviert den `complain`-Modus für das AppArmor-Profil des angegebenen Programms. Regelverstöße werden damit zwar protokolliert, aber nicht geahndet, d. h., das Programm läuft ungestört weiter.

`aa-complain` steht wie alle anderen `aa-xxx`-Kommandos nur bei Distributionen zur Verfügung, die das Sicherheitssystem AppArmor verwenden. Gegenwärtig sind dies z. B. (open)SUSE und Ubuntu.

### Beispiel

Das folgende Kommando bewirkt, dass das Programm `dovecot` zwar weiterhin überwacht wird, Regelverstöße aber lediglich in einer Logging-Datei festgehalten werden:

```
root# aa-complain /usr/sbin/dovecot
Setting /usr/sbin/dovecot to complain mode.
```

### **aa-disable** programm

`aa-disable` deaktiviert das AppArmor-Profil für das angegebene Programm. Zur Reaktivierung verwenden Sie `aa-enforce` oder `aa-complain`.

### **aa-enforce** programm

Das Kommando `aa-complain` aktiviert den `enforce`-Modus für das AppArmor-Profil des angegebenen Programms. AppArmor unterbindet damit Datei- oder Netzwerkzugriffe, die durch AppArmor-Regeln verboten sind.

### **aa-status** [option]

`aa-status` liefert eine Zusammenfassung über den Zustand des AppArmor-Systems. Durch die Angabe genau einer Option aus `--complaining`, `--enabled`, `--enforced` oder `--profiled` kann die Ausgabe auf eine Detailinformation verkürzt werden.

## Beispiel

Auf dem Testrechner ist AppArmor aktiv. Fünf Regelprofile sind geladen, aber nur eines überwacht ein tatsächlich laufendes Programm.

```
root# aa-status
apparmor module is loaded.
5 profiles are loaded.
5 profiles are in enforce mode.
  /sbin/dhclient
  /usr/lib/NetworkManager/nm-dhcp-client.action
  /usr/lib/connman/scripts/dhclient-script
  /usr/sbin/mysqld
  /usr/sbin/tcpdump
0 profiles are in complain mode.
1 processes have profiles defined.
1 processes are in enforce mode.
  /usr/sbin/mysqld (7188)
0 processes are in complain mode.
0 processes are unconfined but have a profile defined.
```

## acpi [optionen]

acpi liefert Informationen zur Energieversorgung des Rechners.

- ▶ -a  
zeigt an, ob der Rechner an das Stromnetz angeschlossen ist.
- ▶ -b  
zeigt den Batteriezustand an.
- ▶ -t  
zeigt die Temperatur des Akkus an.
- ▶ -V  
zeigt alle verfügbaren Informationen an.

## add-apt-repository ppa:name

Das Ubuntu-spezifische Kommando `add-apt-repository` aus dem Paket `software-properties-common` richtet eine neue Paketquelle für ein *Personal Package Archive* (PPA) ein. PPAs sind nichtoffizielle Paketquellen von Ubuntu-Entwicklern, in denen oft aktuelle Versionen oder Entwicklerversionen populärer Programme zu finden sind.

### Beispiel

Das folgende Kommando richtet eine Paketquelle für die jeweils aktuellste Version von LibreOffice ein:

```
root# add-apt-repository ppa:libreoffice/ppa
```

```
addgroup [optionen] name
```

`addgroup` richtet auf Debian- und Ubuntu-Systemen eine neue Gruppe ein und berücksichtigt dabei die Einstellungen aus /etc/adduser.conf. Bei Fedora und Red Hat ist `addgroup` ein Link auf das Kommando groupadd mit eigener Syntax.

► `--gid n`

weist der Gruppe die angegebene GID-Nummer zu (Group Identification).

```
adduser [optionen] name
```

```
adduser name group
```

`adduser` richtet auf Debian- und Ubuntu-Systemen einen neuen Benutzer ein und berücksichtigt dabei die Einstellungen aus /etc/adduser.conf. Insbesondere wird für jeden Benutzer auch eine gleichnamige Gruppe erzeugt und dem neuen Benutzer zugeordnet. In der zweiten Syntaxvariante fügt `adduser` den bereits vorhandenen Benutzer einer zusätzlichen Gruppe hinzu.

Bei Fedora und Red Hat ist `adduser` ein Link auf das Kommando useradd, sodass dort eine andere Syntax gilt. Unter (open)SUSE steht `adduser` nicht zur Verfügung, Sie müssen stattdessen useradd verwenden.

► `--disabled-login`

verzichtet auf die Passwort-Frage. Ein Login ist unmöglich, bis das Passwort eingestellt wird.

► `--gecos 'vollständiger Name'`

verzichtet auf die Fragen nach dem vollständigen Namen, dem (Büro-)Raum, den Telefonnummern und anderen Informationen.

► `--group`

richtet nur eine neue Gruppe ein (keinen Benutzer). `adduser --group` entspricht `addgroup`.

► `--system`

richtet einen System-Benutzer ein. Dabei wird die UID aus dem Bereich für System-Benutzer gewählt (in der Regel 100–999) und der Login blockiert (Shell `/bin/false`). Es kann kein Passwort angegeben werden.

► `--uid n`

weist dem Benutzer die angegebene UID-Nummer zu (User Identification).

### Beispiel

Das folgende Kommando richtet auf einem Debian/Ubuntu-System den neuen Benutzer `kofler` ein und fügt diesen anschließend der Gruppe `libvirt` hinzu:

```
root# adduser --gecos 'Michael Kofler' kofler
root# adduser kofler libvirt
```

```
alias abkürzung='kommando'
```

Das Shell-Kommando `alias` definiert eine neue Abkürzung bzw. zeigt eine vorhandene Abkürzung an. Wenn `alias` ohne weitere Parameter verwendet wird, werden alle definierten Abkürzungen angezeigt.

Unter Fedora und Red Hat gibt es für `root` einige vordefinierte Aliase. Sie werden beim Login aus der Datei `/root/.bashrc` gelesen.

### Beispiel

Das folgende Kommando definiert die Abkürzung `ll` für das Kommando `ls -l`:

```
user$ alias ll='ls -l'
```

```
alien [optionen] paketdatei
```

`alien` wandelt die angegebene Paketdatei in ein anderes Format um. Das gewünschte Format wird durch die Optionen `--to-deb` (Debian), `--to-rpm` (RPM) oder `--to-tgz` (tar-Archiv) angegeben. `alien` muss von `root` ausgeführt werden, damit die Besitzer und Zugriffsrechte der neuen Pakete richtig eingestellt werden.

`alien` funktioniert nur bei einfachen Paketen problemlos. Wenn ein Paket dagegen Installations-Skripts oder andere spezifische Eigenheiten des jeweiligen Paketformats nutzt, wird die Installation des zuvor umgewandelten Pakets voraussichtlich scheitern.

```
alsactl [optionen] [kommando] [card/id/device]
```

Mit `alsactl` können Sie den Zustand des Audio-Systems anzeigen, speichern und wiederherstellen. Außerdem bietet das Kommando einige fortgeschrittene Steuerungsmöglichkeiten, die im einfacheren Kommando `amixer` nicht zur Verfügung stehen.

`alsactl` gilt generell für alle Audio-Geräte, es sei denn, Sie geben beim Aufruf des Kommandos explizit die Nummer oder den Device-Namen des Audio-Geräts an (siehe auch die Datei `/proc/asound/cards`).

- ▶ `-b`  
führt `alsactl` im Hintergrund aus. Diese Option hilft dem Init-System dabei, `alsactl` als Hintergrunddienst zu starten.
- ▶ `-f datei`  
gibt an, welche Datei `alsactl` auslesen bzw. zum Speichern verwenden soll (standardmäßig `/var/lib/alsa/asound.state`).
- ▶ `-s`  
verwendet den Syslog-Dienst zur Protokollierung von Nachrichten, Warnungen und Fehlern.

Die wichtigsten Kommandos von `alsactl` gehen aus der nachfolgenden Aufzählung hervor. Es gibt einige weitere Kommandos, die speziell für den Einsatz als Hintergrunddienst gedacht sind und für den interaktiven Aufruf bzw. für Scripts selten von Interesse sind.

- ▶ `init`  
initialisiert das Audio-System mit Defaulteinstellungen.
- ▶ `monitor`  
verfolgt alle Änderungen des Audio-Systems und zeigt diese an, bis das Kommando mit `[Strg]+[C]` beendet wird.
- ▶ `restore`  
stellt einen zuvor gespeicherten Zustand des Audio-Systems wieder her. Die Daten werden aus `/var/lib/alsa/asound.state` oder aus der durch `-f` angegebenen Datei gelesen.
- ▶ `store`  
speichert den Zustand des Audio-Systems in der Textdatei `/var/lib/alsa/asound.state` oder in einer anderen, durch `-f` angegebenen Datei.

**alsamixer** [optionen]

alsamixer ist ein interaktives Kommando, in dem Sie durch Funktions- und Cursor-tasten die Lautstärke bzw. den Aufnahmepegel der Audio-Kanäle einstellen können. `[Esc]` beendet das Programm. alsamixer ist nicht zur Script-Programmierung geeignet. Dazu verwenden Sie amixer.

▶ `-c n`

wählt die gewünschte Audio-Karte aus, wenn es mehrere gibt. Die Audio-Karten sind durchnummeriert, beginnend mit 0. Werfen Sie gegebenenfalls einen Blick in die Datei `/proc/asound/cards`, die alle Audio-Karten aufzählt.

▶ `-D device`

wählt das Audio-System aus. Beispielsweise können Sie mit `-D pulse` Einstellungen des PulseAudio-Systems verändern. Ohne die Option steuert alsamixer direkt das ALSA-System.

▶ `-V playback|capture|all`

bestimmt, welche Regler anfänglich angezeigt werden sollen: Die zur Audio-Wiedergabe, die zur Aufnahme oder alle. Wenn alsamixer läuft, können Sie zwischen diesen Darstellungsformen mit `[F3]`, `[F4]` und `[F5]` wechseln.

**alternatives** [optionen] kommando

alternatives ist die Red-Hat- bzw. Fedora-Variante von update-alternatives. Die Syntax der beiden Kommandos ist weitgehend identisch und wird beim Kommando update-alternatives beschrieben.

**amixer** [optionen] [kommando]

Mit dem Kommando amixer können Sie durch Optionen die Lautstärke bzw. Pegel der Kanäle des Audio-Systems verändern. Wenn Sie die Lautstärke oder andere Einstellungen interaktiv regeln möchten, bietet sich dazu das Kommando alsamixer mit einer einfachen, textbasierten Benutzeroberfläche an.

▶ `-c n`

wählt die gewünschte Audio-Karte aus. Die Audio-Karten sind durchnummeriert, beginnend mit 0. Werfen Sie gegebenenfalls einen Blick in die Datei `/proc/asound/cards`, die alle Audio-Karten aufzählt. Wenn die Option fehlt, bearbeitet amixer das Default-Audio-Device.

► `-D device`

wählt das Audio-System aus (z. B. `-D pulse` für das PulseAudio-System). Ohne die Option steuert `amixer` direkt das ALSA-System.

► `-q`

verzichtet darauf, nach der Veränderung von Einstellungen den neuen Zustand auszugeben.

► `-s`

liest die auszuführenden Kommandos aus der Standardeingabe. Dabei werden ausschließlich die Kommandos `sset` und `cset` berücksichtigt.

Die folgende Aufzählung beschreibt die wichtigsten `amixer`-Kommandos:

► `cget id`

zeigt den Zustand eines Schalters an. Die Bezeichnung des Schalters kann durch einen oder mehrere Parameter erfolgen, z. B. in der Form `numid=3` oder `iface=MIXER,name='Headphone Playback Switch'`. Eine Liste aller Regler und Schalter samt entsprechender ID-Daten erstellt `amixer controls`.

► `controls|scontents|scontrols`

listet alle Regler bzw. Schalter in einer unterschiedlich detaillierten Darstellung auf und gibt deren aktuellen Zustand an.

► `cset id parameter`

verändert den Zustand eines Schalters. Beispielsweise schaltet `amixer cset numid=3 1` auf einem Raspberry Pi den Audio-Ausgang ein.

► `info`

zeigt eine kurze Beschreibung der Audio-Karte an.

► `sget 'name',n`

liefert den Zustand des angegebenen Reglers. Regler werden durch eine Kombination aus Zeichenkette und Zahl beschrieben, z. B. `'Master',0`. Die Zahl ist zumeist 0, es sei denn, eine Audio-Karte bietet mehrere gleichnamige Regler an, die dann durchnummeriert werden.

► `sset 'name',n parameter`

verändert den Zustand des Reglers. Im einfachsten Fall geben Sie als Parameter einen Prozentwert an, z. B. 50%. Falls es sich um einen Stereo-Regler handelt, können Sie die beiden Kanäle durch ein Zahlenpaar getrennt einstellen, z. B. mit



40%, 55%. Alternativ können viele Regler mit dem Suffix dB auch in der Einheit Dezibel eingestellt werden, z. B. -12.5dB.

Durch nachgestellte Plus- oder Minus-Zeichen können Sie Regler auch relativ verändern. 5%+ erhöht die Lautstärke oder den Aufnahmepegel um fünf Prozentpunkte, 5%- reduziert Lautstärke bzw. Pegel um fünf Prozentpunkte.

Mit `mute`, `unmute` und `toggle` verändern Sie den Mute-Modus. `cap` aktiviert die Aufnahme (*Capture*), `nocap` deaktiviert diese Funktion.

Falls der Regler mehrere Kanäle unterstützt, können Sie diese mit den zusätzlichen Parametern `front`, `rear`, `center` und `woofer` auswählen.

### Beispiel

Auf dem Testrechner ist das dritte Audio-Gerät (-c 2, die Nummerierung beginnt mit 0) ein USB-Headset. Es gibt nur zwei *Simple Controls* zur Steuerung der Kopfhörer und des Mikrofons. `amixer sset` stellt nun die Kopfhörer auf eine mittlere Lautstärke.

```
user$ amixer -c 2 scontrols
Simple mixer control 'Headphone',0
Simple mixer control 'Mic',0
user$ amixer -c 2 sset 'Headphone',0 50% unmute
Simple mixer control 'Headphone',0
  Capabilities: pvolume pswitch pswitch-joined
  Playback channels: Front Left - Front Right
  Limits: Playback 0 - 41
  Mono:
  Front Left: Playback 21 [51%] [-20.00dB] [on]
  Front Right: Playback 21 [51%] [-20.00dB] [on]
```

### apropos thema

`apropos` liefert eine Liste aller `man`-Texte, die Informationen zum angegebenen Thema enthalten. Wenn `apropos` nicht funktioniert, fehlen wahrscheinlich die zugrunde liegenden Datenbanken, die mit `mandb` bzw. mit `makewhatis` erzeugt werden können.

### Beispiel

`apropos editor` liefert eine Liste diverser installierter Edit-Kommandos:

```
user$ apropos editor
ed (1)      - line-oriented text editor
editor (1)  - Vi IMproved, a programmers text editor
ex (1)      - Vi IMproved, a programmers text editor
```

```
jmacs (1) - Joe's Own Editor
joe (1)   - Joe's Own Editor
...
```

### apt-cache kommando

Das Debian- und Ubuntu-spezifische Kommando `apt-cache` liefert Informationen über verfügbare bzw. bereits installierte Pakete.

- ▶ `policy`  
zeigt eine detaillierte Liste aller Paketquellen.
- ▶ `policy paketname`  
zeigt an, aus welcher Paketquelle das gegenwärtig installierte Paket stammt und welche Paketquellen gegebenenfalls alternative Versionen des Pakets zur Verfügung stellen.
- ▶ `search muster`  
liefert eine Liste aller verfügbaren Pakete (egal, ob sie bereits installiert sind oder nicht), in deren Paketnamen oder Paketkurzbeschreibung der Suchtext *muster* vorkommt. Die zusätzliche Option `--names-only` schränkt die Suche auf den Paketnamen ein.
- ▶ `show paketname`  
liefert eine kurze Paketbeschreibung. Das funktioniert auch für nicht installierte Pakete, sofern sich die Paketbeschreibung im Cache befindet.
- ▶ `showpkg paketname`  
zeigt die Abhängigkeitsinformationen für das Paket an.
- ▶ `stats`  
liefert eine Statistik über die Anzahl der installierten und verfügbaren Pakete.

`apt-cache` ist nicht in der Lage, detaillierte Informationen über den genauen Inhalt eines Pakets zu geben bzw. eine Zuordnung zwischen einer Datei und dem dazugehörenden Paket herzustellen. Wenn Sie an diesen Informationen interessiert sind, müssen Sie `dpkg` einsetzen.

### Beispiele

Das folgende Kommando liefert eine sortierte Liste aller Pakete, deren Namen `gimp` enthalten:

```
root# apt-cache search --names-only gimp | gimp
```

Das zweite Beispiel zeigt, welche Version des Pakets `openjdk-7-jre` installiert ist und welche alternativen Versionen es gibt:

```
user$ apt-cache policy openjdk-7-jre
openjdk-7-jre:
  Installiert:          7u85-2.6.1-5ubuntu0.15.10.1
  Installationskandidat: 7u85-2.6.1-5ubuntu0.15.10.1
  Versionstabelle:
*** 7u85-2.6.1-5ubuntu0.15.10.1 0
    500 http://at.archive.ubuntu.com/ubuntu/ wily-updates/main amd64 Packages
    500 http://security.ubuntu.com/ubuntu/ wily-security/main amd64 Packages
    100 /var/lib/dpkg/status
  7u85-2.6.1-5ubuntu0.14.04.1 0
    500 http://security.ubuntu.com/ubuntu/ trusty-security/main amd64 Packages
  7u85-2.6.1-5 0
    500 http://at.archive.ubuntu.com/ubuntu/ wily/main amd64 Packages
```

**apt-cdrom** [optionen] add

Das Debian- und Ubuntu-spezifische Kommando `apt-cdrom` ermittelt alle auf der CD/DVD verfügbaren Pakete und trägt sie in eine APT-Cache-Datei ein. Gleichzeitig wird die CD/DVD als zusätzliche Paketquelle in `/etc/apt/sources.list` eingefügt.

Damit `apt-cdrom` richtig funktioniert, muss die CD/DVD Metadaten für das APT-Paketverwaltungssystem enthalten. Wenn `apt-cdrom` die CD/DVD nicht findet, geben Sie deren `mount`-Verzeichnis mit der Option `-d` an.

**apt-get** [optionen] kommando

Das Debian- und Ubuntu-spezifische Kommando `apt-get` installiert, aktualisiert und entfernt Pakete. `apt-get` lädt dabei die Pakete von den in `/etc/apt/sources.list` definierten Paketquellen herunter. Sämtliche Paketverwaltungsfunktionen werden in Form von Kommandos durchgeführt, z. B. `apt-get install paketname`. Die folgende Aufzählung beschreibt die wichtigsten Kommandos:

► `autoclean`

entfernt nur solche Pakete aus dem Paket-Cache, zu denen bereits eine neuere Version verfügbar ist.

- ▶ `autoremove`  
deinstalliert alle nicht mehr benötigten Pakete, die aufgrund von Abhängigkeiten installiert wurden.
- ▶ `check`  
aktualisiert den Cache aller installierten Pakete und stellt eventuell vorhandene Paketkonflikte und ungelöste Abhängigkeiten fest. Das ist nur erforderlich, wenn Pakete ohne APT (de)installiert wurden und in der Folge Konflikte aufgetreten sind.
- ▶ `clean`  
entfernt alle heruntergeladenen Pakete aus dem Paket-Cache.
- ▶ `dist-upgrade`  
funktioniert ähnlich wie `upgrade`, installiert bei Bedarf aber auch neue Pakete.
- ▶ `install name1 name2 ...`  
sucht die Pakete `name1`, `name2` etc. auf allen APT-Paketquellen, lädt sie herunter und installiert sie. Gegebenenfalls werden auch weitere Pakete geladen und installiert oder aktualisiert, um Paketabhängigkeiten zu erfüllen. Bei `name1`, `name2` etc. darf es sich auch um lokale Debian-Dateien handeln. Damit werden diese Pakete installiert, wobei zur Auflösung der Paketabhängigkeiten weiterhin alle APT-Paketquellen genutzt werden.
- ▶ `remove name1 name 2 ...`  
deinstalliert die angegebenen Pakete.
- ▶ `source name`  
installiert den Quellcode des Pakets in das aktuelle Verzeichnis.
- ▶ `update`  
aktualisiert die Paketlisten der in `sources.list` angegebenen Archive. Dabei werden nur die Metadaten der Paketquellen gelesen und in APT-Cache-Dateien eingetragen. Es werden aber keine Pakete heruntergeladen oder aktualisiert! Der einzige Zweck dieses Kommandos ist, dass `apt-get` weiß, welche Pakete im Internet zur Verfügung stehen. Das Kommando sollte vor jedem anderen `apt`-Kommando ausgeführt werden.  
  
Hinweis: Wenn Sie nur ein Paket aktualisieren möchten (nicht alle Pakete), müssen Sie `apt-get install name` ausführen, nicht `apt-get update name`!

- ▶ `upgrade`

aktualisiert alle installierten Pakete, soweit in den Paketquellen neuere Versionen zur Verfügung stehen.

Das Detailverhalten bei der Ausführung der Kommandos wird durch Optionen gesteuert:

- ▶ `-d` bzw. `--download-only`

lädt die Pakete nur in das Verzeichnis `/var/cache/apt/archives` herunter, installiert sie aber nicht.

- ▶ `--no-install-recommends`

verzichtet auf die Installation empfohlener Pakete.

- ▶ `-s` bzw. `--simulate`

simuliert die Installation, führt aber keine tatsächlichen Veränderungen durch.

- ▶ `-y` bzw. `--assume-yes`

beantwortet alle Fragen mit `yes` und ermöglicht so die Verwendung des Kommandos in einem Script.

## Beispiel

Die folgenden Kommandos zeigen die Anwendung von `apt-get`:

```
root# apt-get update          (Paketquellen aktualisieren)
root# apt-get dist-upgrade    (alle installierten Pakete aktualisieren)
root# apt-get install jmacs   (das Paket jmacs installieren)
```

## apt-key kommando

Das Debian- und Ubuntu-spezifische Kommando `apt-key` verwaltet die öffentlichen Schlüssel von APT-Paketquellen. `apt-key` kennt unter anderem die folgenden Befehle:

- ▶ `add schlüsseldatei.gpg`

fügt den Schlüssel aus der angegebenen Datei zur Liste der APT-Schlüssel hinzu.

- ▶ `del id`

löscht den anhand der ID-Nummer spezifizierten Schlüssel aus der Liste der APT-Schlüssel.

- ▶ `list`

liefert eine Liste aller dem APT-System bekannten Schlüssel.

## Beispiel

Die beiden folgenden Kommandos laden den Schlüssel für die VirtualBox-Paketquelle von Oracle herunter und fügen ihn zur Liste der APT-Schlüssel hinzu:

```
root# wget -q https://download.virtualbox.org/virtualbox/debian/oracle_vbox.asc
root# apt-key add oracle_vbox.asc
```

**aptitude** [optionen] [kommando]

Das Debian-spezifische Kommando `aptitude` installiert, aktualisiert und entfernt Pakete und greift dabei wie `apt-get` auf die APT-Infrastruktur zurück. Der Vorteil im Vergleich zu `apt-get` besteht darin, dass sich `aptitude` bei der Installation abhängige Pakete merkt und diese bei der Deinstallation automatisch wieder entfernt.

Sämtliche Paketverwaltungsfunktionen werden in Form von Kommandos durchgeführt (z.B. `aptitude install paketname`). Die elementaren Kommandos stimmen mit denen von `apt-get` überein und sind bei diesem Kommando beschrieben. Die Kommandos `check` und `autoremove` sind in `aptitude` allerdings nicht verfügbar.

Alternativ können Sie das Programm auch mit einer Text-Benutzeroberfläche in einer Konsole nutzen, indem Sie das Programm einfach ohne weitere Parameter starten. Zur Menüauswahl verwenden Sie die Tastenkombination `[Strg]+[T]`. Wirklich intuitiv ist `aptitude` aber trotz des Menüs nicht zu bedienen; die meisten Anwender verwenden `aptitude` daher wie `apt-get` einfach zum Ausführen einzelner Kommandos.

## Beispiel

Die beiden folgenden Kommandos installieren zuerst das Paket `mysql-server` und entfernen es dann wieder. Bemerkenswert ist, dass beim zweiten Kommando die zahlreichen zusammen mit `mysql-server` installierten abhängigen Pakete ebenfalls wieder entfernt werden.

```
root# aptitude install mysql-server
```

Die folgenden NEUEN Pakete werden zusätzlich installiert:

```
libaio1a libdbd-mysql-perla libdbi-perla libhtml-template-perla
libmysqlclient18a libnet-daemon-perla libplrpc-perla
mysql-client-5.5a mysql-commona mysql-server mysql-server-5.5a
mysql-server-core-5.5a
```

0 Pakete aktualisiert, 12 zusätzlich installiert, 0 werden entfernt

```
root# aptitude remove mysql-server
```

Die folgenden Pakete werden ENTFERNT:

```
libaio1u ... mysql-server-core-5.5u
```

0 Pakete aktualisiert, 0 zusätzlich installiert, 12 werden entfernt

**at** [optionen] zeitpunkt

Mit **at** geben Sie an, dass ein oder mehrere Kommandos zu einem späteren Zeitpunkt ausgeführt werden sollen. Die Kommandos werden im Anschluss an den **at**-Befehl interaktiv eingegeben; die Eingabe endet mit `[Strg]+[D]`.

Sofern der Job eine Standardausgabe produziert, wird die Ausgabe nach der Erledigung des Auftrags per E-Mail an den Benutzer versendet, der **at** ausgeführt hat. Das setzt voraus, dass auf dem Rechner `/usr/sbin/sendmail` installiert ist.

## Zeitangabe

Für die Zeitangabe gibt es eine Menge Varianten, von denen im Folgenden die wichtigsten anhand von Beispielen vorgestellt werden:

► `3:45 [tomorrow]`

führt die Kommandos um 3:45 Uhr aus – und zwar heute, wenn es früher ist, oder morgen, wenn 3:45 Uhr bereits vorbei ist. Mit dem nachgestellten `tomorrow` können Sie explizit einen Zeitpunkt am nächsten Tag angeben.

► `[16:30] [2016-]12-31`

führt die Kommandos am angegebenen Tag und zum angegebenen Zeitpunkt aus. Die Jahres- und die Zeitangabe sind optional. Wenn diese Daten fehlen, wird das aktuelle Jahr und die aktuelle Zeit verwendet. `at 11-30` bedeutet also: am 30. November dieses Jahres zur gleichen Uhrzeit, zu der das **at**-Kommando ausgeführt wurde.

► `noon | teatime | midnight`

entspricht den Zeitangaben 12:00 Uhr, 16:00 Uhr und 0:00 Uhr.

► `now`

führt die Kommandos sofort aus. `now` wird oft mit `+ xxx` kombiniert.

► `... + 2 days`

fügt der vorangegangenen Zeitangabe zwei Tage hinzu. Zulässige Zeiteinheiten sind `minutes`, `hours`, `days` und `weeks`. Die grammatikalisch fragwürdige Angabe `at now + 1 hours` bedeutet somit: in einer Stunde. (Zeiteinheiten sind immer im Plural anzugeben.)

### Optionen

- ▶ `-b`  
führt das Kommando aus, sobald das System eine geringe Auslastung hat. Die Zeitangabe entfällt bei dieser Option. `at -b` entspricht dem Kommando batch.
- ▶ `-c n`  
gibt an, welche Kommandos durch den Job mit der angegebenen Nummer ausgeführt werden. Die Jobnummern können Sie mit atq ermitteln.
- ▶ `-d` oder `-r n`  
entfernt den durch seine Nummer angegebenen Job aus der Liste der vorgemerkten Aufträge. `at -d` bzw. `at -r` entspricht atrm.
- ▶ `-f datei`  
liest die auszuführenden Kommandos aus der angegebenen Datei, nicht aus der Standardeingabe.
- ▶ `-l`  
listet alle Jobs auf. `at -l` entspricht dem Kommando atq.
- ▶ `-m`  
Immer eine Mail senden, auch wenn der Job keine Ausgabe liefert.
- ▶ `-M`  
Keine Mail nach Vollendung des Jobs senden.
- ▶ `-qx`  
ordnet den Job der Warteschlange `x` zu. Dabei ist `x` ein Klein- oder Großbuchstabe. Die Buchstaben `a` und `b` werden für gewöhnliche `at`-Jobs sowie für batch-Jobs verwendet. Je höher der Warteschlangenbuchstabe gewählt wird, desto höher ist der `nice`-Wert, mit dem der Job ausgeführt wird. `-qz` eignet sich also für Jobs mit sehr geringer Dringlichkeit.
- ▶ `-t YYYYMMDDhhmm`  
führt den Job zur angegebenen Zeit aus. Die Option `-t` ist eine Alternative zu den anderen Formen der Zeitangabe.

### Beispiel

Es wurden Änderungen an der Konfiguration des Webserver durchgeführt, die erst mit einem Neustart wirksam werden. Um zu vermeiden, dass gerade aktive Benutzer



der Website Session-Daten verlieren, soll der Webserver aber nicht sofort, sondern erst am nächsten Tag um 1 Uhr morgens neu gestartet werden. Der Service-Name `apache2` gilt dabei für Debian und Ubuntu. Unter Fedora und RHEL geben Sie stattdessen `httpd` an.

```
root# at 1:00
service apache2 restart
<Strg>+<D>
```

### aticonfig [optionen]

`aticonfig` hilft dabei, den proprietären AMD/ATI-Grafiktreiber zu konfigurieren. Das Kommando steht nur zur Verfügung, wenn dieser Treiber installiert ist.

`aticonfig` muss mit `root`-Rechten ausgeführt werden. Idealerweise sollte während der Ausführung kein X-Server laufen. Gegebenenfalls müssen Sie also den Display-Manager stoppen (z.B. mit `service lightdm/gdm stop`) und sich in einer Textkonsole einloggen. In der Praxis ist es zumeist sinnvoll, zuerst mit `aticonfig --initial` eine Grundkonfiguration herzustellen. Weitere Änderungen an der Konfiguration führen Sie dann am besten mit der grafischen Benutzeroberfläche `amdccle` aus.

`aticonfig` speichert die Einstellungen in `/etc/X11/xorg.conf`. Wenn die Datei bereits existiert, wird sie überschrieben. Die ursprüngliche Datei wird unter dem Namen `/etc/X11/xorg.conf.fglrx-nnn` archiviert.

#### ► `--initial[=dual-head]`

richtet `xorg.conf` in der Default-Konfiguration ein (optional für den Betrieb mit zwei Monitoren). Wenn es mehr als zwei Monitore gibt, geben Sie außerdem die Option `--heads=n` an.

#### ► `--list-adapters`

listet die verfügbaren Grafikkarten auf. Die Option ist nur relevant, wenn in Ihrem Rechner mehrere Grafikkarten eingebaut sind. In diesem Fall können Sie weitere Kommandos mit `--adapter=n` explizit für eine bestimmte Grafikkarte ausführen.

#### ► `--resolution=n, WxH`

gibt die gewünschte Bildschirmauflösung für den Bildschirm mit der Nummer *n* an (z. B. 0, 1920x1080 für den ersten Monitor).

#### ► `--screen-layout=left|right|above|below`

gibt an, wie der zweite Bildschirm relativ zum ersten positioniert ist.

Zu `atconfig` existiert keine `man`-Seite. Die gesamte Syntax des Kommandos können Sie mit `atconfig --help` nachlesen.

### `atq`

`atq` listet alle Jobs auf, die mit `at` für die Ausführung zu einem zukünftigen Zeitpunkt eingerichtet wurden. Wenn `atq` von `root` ausgeführt wird, zeigt es die Jobs aller Benutzer an, andernfalls nur eigene Jobs.

### `atrm n`

`atrm` löscht den durch seine Nummer angegebenen Job. Die Nummern aller Jobs, die für eine zukünftige Ausführung eingerichtet wurden, ermitteln Sie mit `atq`.

### `attr [optionen] dateien`

`attr` ermittelt bzw. verändert die erweiterten Zugriffsattribute der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem EAs (*Extended Attributes*) unterstützt. Bei `ext3/ext4`-Dateisystemen muss dazu die `mount`-Option `user_xattr` verwendet werden.

Anstelle von `attr` sollten Sie die Kommandos `getfattr` bzw. `setfattr` vorziehen. `attr` steht nur aus Kompatibilitätsgründen zu IRIX SGI zur Verfügung.

### `avconv [inopts] [-i infile] [outopts] outfile`

Das Kommando `avconv` aus dem Paket `libav-tools` ersetzt auf manchen Distributionen das populäre Kommando `ffmpeg`. `avconv` ist ein Fork von `ffmpeg`. Die Syntax beider Kommandos ist nahezu identisch und wird bei `ffmpeg` beschrieben.

### `awk [optionen] 'code' textdateien` `awk [optionen] -f codedatei textdateien`

`awk` ist kein simples Kommando, sondern eine eigene Programmiersprache, die bei der Verarbeitung und Auswertung strukturierter Texte hilft. Sie können damit z. B. eine Textdatei nach Schlüsselwörtern durchsuchen und aus dem nachfolgenden Textabschnitt eine Textspalte extrahieren. Den Programmcode geben Sie wahlweise direkt ein oder lesen ihn mit der Option `-f` aus einer Codedatei. `awk` verarbeitet dann

alle als Parameter übergebenen Textdateien zeilenweise und schreibt das Ergebnis in die Standardausgabe.

Der von `awk` zu verarbeitende Programmcode folgt dieser vereinfachten Syntax:

```
/suchmuster1/ {aktionen}
/suchmuster2/ {weitere aktionen}
bedingung     {noch mehr aktionen}
```

Für den Aufbau des Suchmusters gelten dieselben Regeln, die beim Kommando `grep` beschrieben sind. Mehrere Aktionen werden durch Semikola getrennt. Wenn das Suchmuster fehlt, gilt die Aktion für jede Zeile der Textdatei. Wenn umgekehrt die Aktion fehlt, wird der durch das Muster erfasste Text einfach ausgegeben.

Eine umfassende Beschreibung von `awk` und seinen Möglichkeiten ist hier aus Platzgründen ausgeschlossen. Stattdessen beschränke ich mich auf einige Beispiele. Wenn Sie sich intensiver in `awk` einarbeiten möchten, bietet sich dieses Tutorial als Einführung an:

<http://www.grymoire.com/Unix/Awk.html>

Der offizielle *User's Guide* geht mehr ins Detail, kann mit fast 500 Seiten Umfang aber Einsteiger abschrecken:

<http://www.gnu.org/software/gawk/manual/gawk.pdf>

## Beispiele

Als Ausgangspunkt für die folgenden Beispiele dient dieser Auszug aus `/etc/passwd`:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
...
```

Das erste `awk`-Beispiel liefert nur die erste und die siebte Spalte aus `/etc/passwd`, also den Login-Namen und die zugeordnete Shell. Die Option `-F` gibt an, dass die Spalten durch Doppelpunkte getrennt sind und nicht durch Leer- und Tabulatorzeichen:

```
user$ awk -F':' '{print $1,$7}' /etc/passwd
root /bin/bash
bin /sbin/nologin
daemon /sbin/nologin
...
```

Das zweite Beispiel ermittelt das Heimatverzeichnis und die Shell von `kofler`:

```
user$ awk -F':' '/kofler/ {print $6; print $7;}' /etc/passwd
/home/kofler
/bin/bash
```

Das dritte Beispiel gibt die Login-Namen aller Benutzer mit einer UID größer gleich 1000 aus:

```
user$ awk -F':' ' $3>=1000 {print $1}' /etc/passwd
nfsnobody
kofler
test
```

Das letzte Beispiel extrahiert aus der Ausgabe von `ls -l` die Dateigröße und den Dateinamen. Beachten Sie aber, dass das nur für Dateinamen funktioniert, die keine Leerzeichen enthalten.

```
user$ ls -l *.txt | awk '{print $5,$9}'
123 datei1.txt
213231 datei2.txt
...
```

### **badblocks** device

`badblocks` führt eine Low-Level-Überprüfung des Datenträgers durch. Das ist nur zweckmäßig, wenn Sie den Verdacht haben, dass die Festplatte defekt sein könnte. Wenn Sie vorhaben, auf der Festplatte(npartition) ein `ext`-Dateisystem einzurichten, ist es besser, die Überprüfung mit `mke2fs -c` im Rahmen der Formatierung durchzuführen.

`badblocks` zerstört normalerweise keine Daten und führt nur einen Read-only-Test durch. Gründlichere Tests können Sie mit den Optionen `-n` oder `-w` durchführen:

#### ► `-n`

führt auch Schreibtests durch, stellt den ursprünglichen Inhalt der Datenblöcke aber wieder her. Die Option kann nicht verwendet werden, wenn der Datenträger bzw. die Partition gerade verwendet wird, also mit `mount` in den Verzeichnisbaum eingebunden ist.

#### ► `-w`

führt Schreibtests mit Bitmustern durch. Vorsicht: Der Inhalt des Datenträgers bzw. der Partition wird dadurch gelöscht!

### **basename** zeichenkette [endung]

`basename` liefert den Dateinamen des übergebenen Pfads. `basename /etc/X11/Xmodmap` führt also zum Ergebnis `Xmodmap`. Wenn als zusätzlicher Parameter eine Dateiendung

angegeben wird, so wird diese Dateiendung (falls vorhanden) aus dem Dateinamen entfernt.

### Beispiel

basename eliminiert den Pfad und die Kennung .jpg. Übrig bleibt white.

```
user$ basename /home/kofler/Bilder/Wallpapers/white.jpg .jpg
white
```

## batch

batch merkt einen Job-Auftrag für die spätere Ausführung vor. Wie bei `at` werden die auszuführenden Kommandos anschließend interaktiv eingegeben. Die Eingabe wird mit `[Strg]+[D]` abgeschlossen. Anders als bei `at` entfällt die Zeitangabe, wann der Job ausgeführt werden soll. Vielmehr wartet der Hintergrunddämon `atd` so lange, bis das System eine geringe Auslastung hat. Die Schwelle ist mit einem `load`-Durchschnitt von 1,5 definiert. Wenn der Job eine Ausgabe liefert, wird diese per Mail an den Benutzer gesendet. Das erfordert, dass auf dem Rechner `/usr/sbin/sendmail` zur Verfügung steht.

batch verarbeitet weder Optionen noch Parameter. Wenn Sie den Jobauftrag aus einer Datei lesen möchten, führen Sie `at -qb -f datei now` aus.

### Beispiel

Mit der Ausführung des Backup-Scripts wird begonnen, sobald die Systemauslastung gering ist:

```
user$ batch
./backup-script
<Strg>+<D>
```

## bg [prozess]

Das Shell-Kommando `bg` setzt einen unterbrochenen Prozess im Hintergrund fort. Wenn keine Prozessnummer angegeben wird, gilt `bg` für den zuletzt mit `[Strg]+[Z]` unterbrochenen Prozess. Andernfalls muss der Prozess durch seinen Namen oder durch die `bash`-interne Jobnummer (nicht durch die PID!) angegeben werden.

## blkid [optionen] [device]

blkid liefert Informationen über das durch den Device-Namen angegebene Block-Device, also z.B. über Festplattenpartitionen, Logical Volumes oder RAID-Geräte. blkid gibt hierfür den Dateisystemtyp, die UUID und den Namen (Label) des Dateisystems an. Wenn die Device-Angabe beim Aufruf von blkid fehlt, liefert blkid diese Informationen für alle aktiven Partitionen und Logical Volumes. Durch die Angabe von Optionen kann gezielt nach bestimmten Dateisystemen gesucht werden:

- ▶ `-k`  
listet alle vom Kernel unterstützten Dateisystemtypen auf.
- ▶ `-L label`  
sucht nach Dateisystemen mit dem angegebenen Namen.
- ▶ `-n fstyp1,fstyp2,fstyp3`  
sucht nach Dateisystemen, die den angegebenen Typen entsprechen.
- ▶ `-t name=wert`  
sucht nach Dateisystemen, die dem Suchkriterium entsprechen. Zulässige Kriterien sind TYPE (Dateisystemtyp), LABEL (Name des Dateisystems), PARTLABEL (Name der Partition), UUID (UUID des Dateisystems) und PARTUUID (UUID der Partition).  
`blkid -t TYPE=ext4` entspricht `blkid -n ext4` und liefert eine Liste aller ext4-Dateisysteme.
- ▶ `-U uuid`  
sucht nach Dateisystemen mit der angegebenen UUID.

### Beispiele

Die Partition `/dev/sda1` enthält ein ext4-Dateisystem mit der UUID 2716...19e4:

```
root# blkid /dev/sda1
/dev/sda1: UUID="27162884-8fe9-4fa9-8b5e-712ab82d19e4" TYPE="ext4"
```

Die Suche nach einer Partition mit dem Label `lvm2` liefert als Ergebnis `/dev/sdb2`:

```
root# blkid -t PARTLABEL=lvm2
/dev/sdb2: PARTLABEL="lvm2" PARTUUID="371cc374-847f-407a-bb5b-7ff015722383"
```

### break [*n*]

Das bash-Kommando `break` bricht in Shell-Scripts eine `for`-, `while`- oder `until`-Schleife vorzeitig ab. Das Shell-Programm wird beim nächsten Kommando nach dem Schleifenende fortgesetzt. Durch die Angabe eines optionalen Zahlenwerts können *n* Schleifenebenen abgebrochen werden.

**btrfs** kommando

btrfs ist das zentrale Administrationskommando für btrfs-Dateisysteme. Sofern keine Doppeldeutigkeiten auftreten, können Sie die im Folgenden beschriebenen Kommandos abkürzen und damit eine Menge Tipparbeit sparen. `btrfs fi sh` entspricht also `btrfs filesystem show`.

Die Administration von btrfs-Dateisystemen ist außergewöhnlich komplex. Im Folgenden sind nur die wichtigsten Kommandos zusammengefasst. Es fehlt hier aber der Platz, die btrfs-Grundlagen im Detail zu beschreiben.

- ▶ `balance start|pause|resume|cancel|status mountverzeichnis`

verteilt die Daten gleichmäßig über alle Devices des Dateisystems. Dieser Vorgang dauert sehr lange und wird im Hintergrund durchgeführt. Er kann mit `btrfs balance pause` unterbrochen, mit `btrfs balance resume` wieder aufgenommen bzw. mit `btrfs balance cancel` ganz gestoppt werden. `btrfs balance status` gibt Auskunft darüber, wie weit der Balance-Vorgang fortgeschritten ist.

- ▶ `device add devicename mountverzeichnis`

fügt einem aktiven btrfs-Dateisystem ein weiteres Device hinzu und vergrößert so dessen Datenpool.

- ▶ `device delete devicename mountverzeichnis`

entfernt ein Device aus dem Datenpool eines btrfs-Dateisystems. Die auf dem Device enthaltenen Daten werden dabei auf andere Devices des Dateisystems übertragen, was bei großen Dateisystemen natürlich lange dauert.

Wenn Sie ein defektes (nicht mehr verfügbares) Device entfernen möchten, geben Sie als Device-Namen `missing` an. Bei RAID-1-Systemen müssen Sie vorher ein Ersatz-Device hinzufügen und `btrfs filesystem balance` ausführen!

- ▶ `filesystem df verzeichnis`

zeigt detaillierte Informationen darüber an, wie viel Platz für Daten, Metadaten und Systemdaten reserviert ist und wie viel Platz davon bereits genutzt wird. Das Kommando liefert allerdings keine Informationen darüber, wie viel nicht reservierter Speicherplatz noch zur Verfügung steht, was die richtige Interpretation der Daten schwierig macht.

- ▶ `filesystem resize [+/-]size mountverzeichnis`

vergrößert oder verkleinert das Dateisystem im laufenden Betrieb. Die neue Größe kann wahlweise absolut oder relativ angegeben werden. Dabei sind die Kürzel `k`, `m` und `g` für `kByte`, `MByte` und `GByte` erlaubt. Als Größenangabe ist auch `max` zuläs-

sig – dann nutzt das Dateisystem in Zukunft die gesamte Größe des zugrunde liegenden Devices. Denken Sie daran, dass Sie die Größe des zugrunde liegenden Devices gegebenenfalls *vor* der Vergrößerung des Dateisystems, aber *nach* einer Verkleinerung anpassen müssen!

Anstelle von `resize` ist es bei `btrfs`-Dateisystemen einfacher, ein zusätzliches Device hinzuzufügen (`btrfs add device`).

► `filesystem show [verzeichnis/device]`

zeigt Basisinformationen zu `btrfs`-Devices an. Das Kommando gibt unter anderem Auskunft darüber, wie groß die Devices sind und wie viel noch nicht für Daten, Metadaten oder Systemdaten reservierter Platz auf den Devices zur Verfügung steht.

► `filesystem sync verzeichnis`

synchronisiert den Inhalt des Dateisystems bzw. eines seiner Verzeichnisse.

► `scrub start|cancel|resume|status verzeichnis/device`

steuert den Scrub-Vorgang. Durch `scrub start` werden alle Dateien des Dateisystems gelesen, und ihre Integrität wird anhand von Prüfsummen kontrolliert. Dieser Prozess läuft im Hintergrund. Sein Fortschritt kann mit `scrub status` überprüft werden. `scrub cancel` unterbricht den Scrub-Vorgang, `scrub resume` setzt ihn wieder fort.

► `subvolume create verzeichnis/name`

erzeugt das Subvolume `name`. Als Verzeichnis muss das `mount`-Verzeichnis oder ein Verzeichnis innerhalb des `btrfs`-Dateisystems angegeben werden. Wenn kein Verzeichnis angegeben wird, verwendet `btrfs` das aktuelle Verzeichnis.

► `subvolume delete verzeichnis/name`

löscht das angegebene Subvolume oder den angegebenen Snapshot. Der belegte Speicher wird allerdings erst nach und nach durch einen Hintergrundprozess freigegeben.

► `subvolume list mountverzeichnis`

listet alle Subvolumes und Snapshots auf. Als Verzeichnis wird das `mount`-Verzeichnis des Dateisystems angegeben. Die Ausgabe kann durch mehrere Optionen beeinflusst werden. Als empfehlenswert hat sich die Kombination `-a -p -t` herausgestellt. Sie bewirkt, dass der Ort des Subvolumes innerhalb der Verzeichnishierarchie exakt angegeben wird, dass zu jedem Subvolume bzw. Snapshot auch das Parent-Subvolume angegeben und das Ergebnis als Tabelle formatiert wird.



- `subvolume set-default id mountverzeichnis`

bestimmt, welches Subvolume bzw. welcher Snapshot in Zukunft standardmäßig von `mount` verwendet wird. Die Volume-ID können Sie mit `btrfs subvolume list` ermitteln.

- `subvolume snapshot [-r] quellverzeichnis zielverzeichnis/name`

erzeugt den Snapshot `name`. Als Quellverzeichnis muss das `mount`-Verzeichnis des Dateisystems oder das Verzeichnis eines vorhandenen Subvolumes oder Snapshots angegeben werden. Das Zielverzeichnis muss sich innerhalb des `btrfs`-Dateisystems befinden. Der Snapshot ist normalerweise veränderlich (d. h., das Dateisystem wird in zwei Zweige gespalten, die beide getrennt voneinander veränderlich sind). Wenn Sie für Backups einen Read-only-Snapshot wünschen, verwenden Sie die Option `-r`.

## Beispiel

`btrfs subvolume list` zeigt die in einem (open)SUSE-System standardmäßig eingerichteten Subvolumes sowie alle Snapshots:

```
root#
btrfs subvolume list / -p -a -t
ID    gen    parent  top level  path
--    ---    -
257   106     5       5          <FS_TREE>/@
260   106     257     257        <FS_TREE>/@/boot/grub2/i386-pc
261   106     257     257        <FS_TREE>/@/boot/grub2/x86_64-efi
262   1305    257     257        <FS_TREE>/@/home
...
268   106     257     257        <FS_TREE>/@/var/lib/libvirt/images
269   106     257     257        <FS_TREE>/@/var/lib/mailman
270   106     257     257        <FS_TREE>/@/var/lib/mariadb
...
258   1158    257     257        <FS_TREE>/@/.snapshots
259   1310    258     258        <FS_TREE>/@/.snapshots/1/snapshot
283   106     258     258        <FS_TREE>/@/.snapshots/2/snapshot
304   669     258     258        <FS_TREE>/@/.snapshots/20/snapshot
...
```

Um im laufenden Betrieb ein Backup der MariaDB-Datenbankdateien zu erstellen, wird vorübergehend ein Read-only-Snapshot für das Subvolume `/var/lib/mariadb` erstellt:

```
root# btrfs subvolume snapshot -r /var/lib/mariadb/ /var/lib/mariadb-backup
Create a readonly snapshot of '/var/lib/mariadb/' in '/var/lib/mariadb-backup'
root# cd /var/lib/mariadb-backup
```

```
root# tar czf /home/kofler/mariadb.tgz . (Backup erstellen)
root# cd (Verzeichnis verlassen)
root# btrfs subvolume delete /var/lib/mariadb-backup
Delete subvolume (no-commit): '/var/lib/mariadb-backup'
```

### Dokumentation

Die man-Seiten zu btrfs sind über mehrere Seiten verteilt, deren Namen sich jeweils aus btrfs, einem Bindestrich und einem Hauptkommando zusammensetzen. Details zu den diversen btrfs-scan-Kommandos können Sie daher mit `man btrfs-scan` nachlesen.

#### **bunzip2** datei.bz2

bunzip2 dekomprimiert eine zuvor mit bzip2 komprimierte Datei. Dabei wird automatisch die Kennung .bz2 im Dateinamen entfernt. bunzip ist ein Link auf bzip2, wobei automatisch die Option -d aktiviert ist.

#### **bzip2** datei

bzip2 komprimiert die angegebenen Dateien und fügt ihnen die Endung .bz2 hinzu. Das Kommando liefert im Regelfall um 20 bis 30 Prozent kleinere Dateien als durch gzip komprimierte Dateien. Allerdings ist die zum Komprimieren erforderliche Rechenzeit bei bzip2 deutlich länger.

##### ► -1 bis -9

gibt an, wie viel Speicherplatz (RAM) der Komprimierungsalgorithmus nutzen darf. Die Grundeinstellung lautet -9 und liefert die besten Ergebnisse. Wenn nur wenig RAM zur Verfügung steht, sollten Sie einen kleineren Wert wählen; allerdings wird dann auch die Komprimierung etwas schlechter.

##### ► -c bzw. --stdout oder --to-stdout

lässt die zu (de)komprimierende Datei unverändert und leitet das Ergebnis auf die Standardausgabe um. Von dort kann es mit > in eine beliebige Datei umgeleitet werden.

##### ► -d bzw. --decompress oder --uncompress

dekomprimiert die angegebene Datei, anstatt sie zu komprimieren (entspricht bunzip2).

## Beispiel

Das erste Kommando komprimiert `datei.eps`. Der neue Dateiname lautet nun `datei.eps.bz2`. Das zweite Kommando stellt die ursprüngliche Datei wieder her.

```
user$ bzip2 datei.eps
user$ bunzip2 datei.eps.bz2
```

```
case ausdruck in
  muster1 ) kommandos;;
  muster2 ) kommandos;;
  ...
esac
```

Das Konstrukt `case` der `bash`-Shell bildet in Scripts Mehrfachverzweigungen, wobei als Kriterium für die Verzweigung eine Zeichenkette angegeben wird – oft eine Variable oder ein Parameter, der dem Shell-Programm übergeben wird. Diese Zeichenkette wird der Reihe nach mit den Mustern verglichen, wobei in diesen Mustern die Jokerzeichen für Dateinamen (`*?[]`) verwendet werden können. In einem `case`-Zweig können auch mehrere durch `|` getrennte Muster angegeben werden. Sobald ein Muster zutrifft, werden die Kommandos ausgeführt, die zwischen der schließenden runden Klammer `)` und den beiden Strichpunkten folgen. Anschließend wird das Programm nach `esac` fortgesetzt.

## Beispiel

Im folgenden Listing wird `case` eingesetzt, um die übergebenen Parameter in Dateinamen und Optionen zu klassifizieren. Innerhalb einer Schleife wird jeder einzelne Parameter mit `case` analysiert. Wenn der Parameter mit einem Bindestrich beginnt, wird der Parameter an das Ende der Variablen `opt` angefügt, andernfalls an das Ende von `dat`.

```
#!/bin/bash
for i do # Schleife für alle übergebenen Parameter
  case "$i" in
    -* ) opt="$opt $i";;
    * ) dat="$dat $i";;
  esac
done # Ende der Schleife
echo "Optionen: $opt"
echo "Dateien: $dat"
```

```
cat [optionen] [dateien]
```

cat zeigt den Inhalt der angegebenen Textdatei an. cat wird häufig auch dazu verwendet, mehrere Dateien zu einer größeren Datei zusammenzusetzen. Dazu muss die Standardausgabe mit > in eine Datei umgeleitet werden (siehe Beispiel). Wenn cat aus der Standardeingabe lesen soll, verzichten Sie auf die Nennung eines Dateinamens oder geben das Zeichen - an.

Bei längeren Texten sollten Sie statt cat das Kommando less verwenden: Damit können Sie zeilen- bzw. seitenweise durch den Text blättern. Zu cat existiert auch die Variante tac. Dieses Kommando gibt die Zeilen der Textdatei in umgekehrter Reihenfolge aus, also die letzte Zeile zuerst.

► -s

reduziert mehrere leere Zeilen auf eine einzige leere Zeile.

► -T

zeigt Tabulatorzeichen in der Form ^I an.




► -v

zeigt nicht druckbare Zeichen in der ^xxx-Schreibweise an.

## Beispiele

Das folgende Kommando setzt die Einzeldateien teil1.tex, teil2.tex etc. zu einer Gesamtdatei total.tex zusammen. Die Einzeldateien werden in alphabetischer Reihenfolge verarbeitet.

```
user$ cat teil*.tex > total.tex
```

Im zweiten Beispiel wird die Standardeingabe in eine neue Datei umgeleitet. Nachdem Sie das Kommando mit  bestätigt haben, werden alle weiteren Eingaben in die neue Datei geschrieben. + beendet die Eingabe. In dieser Form kann cat also dazu verwendet werden, um ohne einen Editor eine neue Textdatei zu erzeugen.

```
root# cat > neuedatei
Zeile 1
Zeile 2
<Strg>+<D>
```

```
cd [verzeichnis]
```

Das Shell-Kommando cd wechselt in das angegebene Verzeichnis. Wenn kein Verzeichnis angegeben wird, wechselt cd in das Heimatverzeichnis. Wenn als Verzeichnis -

angegeben wird, wechselt `cd` in das zuletzt gültige Verzeichnis. `pwd` zeigt den Pfad des gerade aktuellen Verzeichnisses an.

```
cdrdao kommando [optionen] toc-datei
```

`cdrdao` schreibt eine CD im Disk-at-once-Modus (DAO). In der Praxis besteht die gebräuchlichste Anwendung von `cdrdao` darin, Audio-CDs zu kopieren.

### Beispiel

Das erste `cdrdao`-Kommando erzeugt die Dateien `data.bin` (Inhalt der CD) und `data.toc` (Inhaltsverzeichnis). Das zweite Kommando schreibt diese Daten auf eine CD.

```
user$ cdrdao read-cd --device /dev/sr0 data.toc
user$ cdrdao write --device /dev/sr0 --buffers 64 data.toc
```

```
cdrecord [optionen] dev=xxx image/audiodateien
```

`cdrecord` schreibt ein ISO-Image oder Audio-Dateien auf einen CD-Rohling. Nach Lizenzstreitigkeiten mit dem `cdrecord`-Autor Jörg Schilling sind die meisten Distributionen auf den `cdrecord`-Fork `wodim` umgestiegen. Zu den Ausnahmen zählt (open)SUSE, wo `cdrecord` standardmäßig installiert ist. Die wichtigsten Optionen zur Steuerung von `cdrecord` sind beim Kommando `wodim` zusammengefasst.

```
cdisk [optionen] device
```

`cdisk` hilft bei der Formatierung von Festplatten und SSDs. Es unterstützt Partitionstabellen in den Formaten MBR und GTP. Die Bedienung ist im Vergleich zu den bekannteren Kommandos `fdisk` und `parted` etwas komfortabler: Die aktuelle Partitionstabelle wird als Menü angezeigt. Mit den Cursortasten kann nun die zu bearbeitende Partition ausgewählt werden. `cdisk` ist nur für die interaktive Verwendung gedacht und ist nicht zur Script-Programmierung geeignet.

```
chac1 [optionen] dateien
```

`chac1` ermittelt bzw. verändert die erweiterten Zugriffsrechte der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem ACLs (Access

*Control Lists*) unterstützt. Bei ext3/ext4-Dateisystemen muss dazu die `mount-Option` `acl` verwendet werden.

Statt `chac1` sollten Sie nach Möglichkeit `getfacl` bzw. `setfacl` einsetzen. `chac1` steht nur aus Kompatibilitätsgründen zu IRIX SGI zur Verfügung.

```
chage [optionen] loginname
```

Mit `chage` stellen Sie ein, wie lange ein Benutzerkonto verwendet werden darf, wann dessen Passwort abläuft und verändert werden muss und wie oft das Passwort verändert werden darf bzw. muss.

► `-d datum`

setzt das Datum fest, an dem das Passwort zuletzt verändert wurde. Das Datum wird in der Schreibweise JJJJ-MM-TT (also z. B. 2016-12-31) angegeben. `-d 0` bewirkt, dass der Benutzer sofort nach dem ersten Login sein Passwort ändern muss. Der Benutzer kann kein Kommando ausführen, bevor das Passwort verändert ist.

► `-E datum`

setzt das Datum fest, an dem das Konto abläuft. Ab diesem Datum ist kein Login mehr möglich. Mit `-E -1` gibt es kein Ablaufdatum. Abgelaufene Konten können mit `passwd -u` wieder aktiviert werden.

► `-I n`

gibt an, wie viele Tage nach Ablauf des Passworts ein Konto deaktiviert wird. `-I -1` verhindert eine Deaktivierung.

► `-l`

zeigt die aktuellen Einstellungen an.

► `-m n`

legt fest, wie viele Tage mindestens verstreichen müssen, bevor ein geändertes Passwort neuerlich verändert werden kann. Die Standardeinstellung 0 bedeutet, dass der Benutzer sein Passwort jederzeit ändern darf.

► `-M n`

legt fest, nach wie vielen Tagen das Passwort spätestens geändert werden muss. `-1` bedeutet, dass das Passwort unbeschränkt gültig bleibt. 183 bedeutet, dass das Passwort zweimal pro Jahr verändert werden muss.

► -W *n*

gibt an, wie viele Tage vor dem Ablauf des Passworts eine Warnung angezeigt wird (standardmäßig 7).

### Beispiel

Mit den folgenden Kommandos richten Sie einen neuen Benutzer ein, legen dessen Startpasswort fest und zwingen ihn, sein Passwort sofort beim ersten Login sowie in Zukunft zumindest einmal jährlich zu ändern. Die Warnung vor dem Ablauf des Passworts erscheint jeweils einen Monat im Voraus.

```
root# useradd peter
root# passwd peter
New password: ******
Retype new password: ******
root# chage -d 0 -M 365 -W 31 peter
```

**chattr** [optionen] +=[aAcCdDeiJsStTu] dateien

In Linux-Dateisystemen (ext2 bis ext4, btrfs, xfs etc.) können zusammen mit jeder Datei neben den Benutzerinformationen (siehe `chmod` und `chown`) einige zusätzliche Attribute gespeichert werden. Diese Attribute mit den Buchstaben-Codes `aAcCdDeiJsStTu` steuern beispielsweise, ob Dateien komprimiert werden sollen, ob Änderungen an Dateien mit dem Copy-on-Write-Verfahren durchgeführt werden sollen etc.

Eine kurze Beschreibung der Attribute gibt man `chattr`. Allerdings werden momentan nur wenige der vorgesehenen Attribute tatsächlich genutzt. Zum Auslesen der Attribute verwenden Sie `lsattr`.

## ► -R

verändert die Attribute rekursiv in allen Dateien des Verzeichnisbaums.

### Beispiel

Eine wichtige Anwendung findet `chattr` bei btrfs-Dateisystemen: Dort können Sie mit `chattr +C` das automatisch aktive Copy-on-Write-Verfahren (COW) deaktivieren bzw. mit `chattr -C` wieder aktivieren. Die Deaktivierung von COW ist erforderlich, um in btrfs-Dateisystemen Image-Dateien von Virtualisierungsprogrammen oder große Datenbankdateien effizient zu speichern.

Dem Kommando übergeben Sie wahlweise einzelne Dateien oder ganze Verzeichnisse. Mit `-R` verändert das Kommando das Attribut rekursiv in Verzeichnisbäumen.

```
root# chattr +C -R /var/lib/mysql
```

Das Attribut gilt allerdings nur für neue Dateien. Um COW bei bereits existierenden Dateien mit einer Größe ungleich 0 Byte zu deaktivieren, müssen Sie die Dateien kopieren:

```
root# mv verz backup
root# mkdir verz
root# chattr +C verz
root# cp -a backup/* verz
root# rm -rf backup
```

```
chcon [optionen] kontext dateien
chcon [optionen] --reference=referenzdatei dateien
chcon [optionen] [-u benutzer] [-r rolle] [-t typ] [-l level] dateien
```

chcon verändert den SELinux-Kontext der angegebenen Dateien. Der neue Kontext kann auf drei Arten angegeben werden: in der Schreibweise `benutzer:rolle:typ:level`, durch die Angabe einer Referenzdatei oder mit den Optionen `-u`, `-r`, `-l` und `-t`. Die dritte Variante hat den Vorteil, dass damit nur ein Teil des gesamten Kontexts verändert werden kann.

Eine manuelle Einstellung des SELinux-Kontexts ist normalerweise dann erforderlich, wenn Sie Dateien oder Verzeichnisse an anderen Orten speichern, als die SELinux-Regeln dies vorsehen – also beispielsweise Apache-Dateien in `/disk2/var/www/html` anstelle von `/var/www/html`. Wenn es Ihnen nur darum geht, den SELinux-Kontext von Dateien zu korrigieren, die sich ohnedies im vorgesehenen Verzeichnis befinden, führt `restorecon` unkomplizierter zum Ziel. Um den SELinux-Kontext vorhandener Dateien festzustellen, verwenden Sie das Kommando `ls` mit der Option `-Z`.

► `-l level`

verändert den Level des SELinux-Kontexts. In den unter RHEL und Fedora üblicherweise geltenden Regeln (Targeted Security) werden Level nicht genutzt. Der Kontext-Level lautet generell `s0` (siehe auch die Datei `/etc/selinux/targeted/setrans.conf`).

► `-r role`

verändert die Rolle des SELinux-Kontexts. SELinux-Rollen legen fest, welcher »Domäne« die Datei zugeordnet wird. SELinux-Domänen bestimmen wiederum, welche Rechte ein Prozess hat.

► `--reference=referenzdatei`

liest den SELinux-Kontext der Referenzdatei und überträgt diesen auf die weiteren angegebenen Dateien.



- ▶ `-R`  
berücksichtigt auch alle Unterverzeichnisse (*recursive*).
- ▶ `-t typ`  
verändert den Typ des SELinux-Kontexts. SELinux-Typen bestimmen die Rechte einer Datei, so wie SELinux-Domänen die Rechte eines Prozesses steuern.
- ▶ `-u benutzer`  
verändert die Benutzerinformation des SELinux-Kontexts. Der SELinux-Benutzer ist eine den SELinux-Regeln bekannte Identität. Gewöhnliche Linux-Benutzer sind durch SELinux-Regeln mit SELinux-Identitäten verbunden (»gemappt«).
- ▶ `-v`  
zeigt die durchgeführten Änderungen an (*verbose*).

### Beispiel

Das folgende Kommando macht HTML-Dateien im Verzeichnis `/disk2/var/www/html` für Apache zugänglich:

```
root# chcon -R system_u:object_r:httpd_sys_content_t:s0 /disk2/var/www/html/*
```

**chgrp** [optionen] gruppe dateien

`chgrp` ändert die Gruppenzugehörigkeit von Dateien. Der Besitzer einer Datei kann diese Datei nur seinen eigenen Gruppen zuordnen. `root` kann beliebige Zuordnungen treffen.

- ▶ `-R` bzw. `--recursive`  
verändert auch die Gruppenzuordnung von Dateien in allen Unterverzeichnissen.

### Beispiel

Das folgende Kommando ordnet alle Dateien im Verzeichnis `/var/www` der Gruppe `www-data` zu. Unter diesem Account läuft unter Debian und Ubuntu der Webserver Apache.

```
root# chgrp -R www-data /var/www/
```

**chkconfig** optionen

`chkconfig` ist ein Red-Hat- bzw. Fedora-spezifisches Kommando zum Einrichten bzw. Löschen von Init-V-Runlevel-Links. Sofern `xinetd` installiert ist, können auch dessen

Dienste mit `chkconfig` gesteuert werden. Das Kommando steht aus Kompatibilitätsgründen vereinzelt auch für andere Distributionen in gleicher oder ähnlicher Syntax zur Verfügung.

Da die meisten Linux-Distributionen `Systemd` als Init-System verwenden (RHEL und CentOS seit Version 7), ist `chkconfig` nur für die ganz wenigen Dienste relevant, die noch durch Init-V-Skripts gestartet werden. Zur Steuerung aller anderen Dienste, die direkt durch `Systemd` gestartet werden, verwenden Sie das Kommando `systemctl`.

► `--add name`

richtet Links auf das Init-V-Skript in den dafür vorgesehenen Standard-Runleveln ein. Diese Information stammt aus den Kommentarzeilen am Beginn des Skripts. Die neuen Links werden erst beim nächsten Runlevel-Wechsel bzw. Neustart berücksichtigt. Das Init-V-Skript wird also nicht gestartet.

► `--del name`

entfernt alle Init-V-Links für das Skript.

► `[--level n] name on|off`

richtet Init-V-Links nur für die angegebenen Runlevel ein bzw. entfernt sie. Wenn die `--level`-Option fehlt (also z.B. `chkconfig name on`), dann werden automatisch die in `/etc/init.d/name` vordefinierten Default-Level berücksichtigt. `chkconfig name on` entspricht somit `chkconfig --add name`, und `chkconfig name off` entspricht `chkconfig --del name`.

► `--list`

liefert eine Liste aller installierten Init-V-Skripts sowie Informationen darüber, in welchen Runleveln die Skripts gestartet werden.

### Beispiel

Das folgende Kommando richtet auf einer alten Distribution mit Init-V-System die Start-Links für den Webserver Apache (Dämon `httpd`) in den Runleveln 3 und 5 ein:

```
root# chkconfig --level 35 httpd on
```

Dabei werden diese zwei Links erzeugt:

```
/etc/rc3.d/S85httpd -> ../init.d/httpd  
/etc/rc5.d/S85httpd -> ../init.d/httpd
```

**chmod** [optionen] änderungen dateien

chmod ändert die neun Zugriffsbits von Dateien. Zusammen mit jeder Datei wird gespeichert, ob der Besitzer (*user*), die Gruppenmitglieder (*group*) und andere Benutzer (*others*) die Datei lesen, schreiben und ausführen dürfen. Die Änderung der Zugriffsbits erfolgt durch die Zeichenkombination *Gruppe +/- Zugriffstyp*, also beispielsweise *g+w*, um allen Gruppenmitgliedern eine Schreiberlaubnis zu geben. Die Gruppe geben Sie durch *u* (*user*), *g* (*group*), *o* (*others*) oder *a* (*all*) an, den Zugriffstyp durch *r* (*read*), *w* (*write*) oder *x* (*execute*).

### Setuid-, setgid- und sticky-Bits

Das setuid-Bit (oft auch suid-Bit genannt) bewirkt, dass Programme so ausgeführt werden, als hätte der Besitzer selbst das Programm gestartet. Wenn der Besitzer eines Programms root ist, dann kann jeder das Programm ausführen, als wäre er selbst root.

Das setgid-Bit hat bei Programmen dieselbe Funktion wie setuid, aber eben für die Gruppenzugehörigkeit. Bei Verzeichnissen bewirkt das setgid-Bit, dass in diesem Verzeichnis neu erzeugte Dateien der Gruppe des Verzeichnisses angehören und nicht, wie sonst üblich, der Gruppe des Benutzers, der die Datei erzeugt.

Das sticky-Bit bewirkt bei Verzeichnissen, in denen alle die Dateien ändern dürfen, dass jeder nur seine eigenen Dateien löschen darf (und nicht auch Dateien anderer Benutzer). Das Bit ist beispielsweise beim /tmp-Verzeichnis gesetzt. In diesem Verzeichnis darf jeder Benutzer temporäre Dateien anlegen. Es muss aber vermieden werden, dass auch jeder Benutzer nach Belieben fremde Dateien umbenennen oder löschen kann.

Um mit chmod die Spezialbits setuid, setgid und sticky zu setzen, sind die folgenden Zeichenkombinationen vorgesehen:

setuid: u+s

setgid: g+s

sticky: +t

Damit setuid wirkt, muss auch das x-Bit für den Besitzer gesetzt sein (u+x).

Damit setgid wirkt, muss auch das x-Bit für die Gruppe gesetzt sein (g+x).

### Oktale Schreibweise

Statt mit Buchstaben kann der Zugriffstyp auch durch eine maximal vierstellige Oktalzahl angegeben werden. Bei den Zugriffsbits ist u, g und o jeweils eine Ziffer zugeordnet. Jede Ziffer ist aus den Werten 4, 2 und 1 für r, w und x zusammengesetzt. 660 bedeutet daher rw-rw----, 777 steht für rwxrwxrwx. Die drei Spezialbits setuid, setgid und sticky haben die Oktalwerte 4000, 2000 und 1000.

Um die Zugriffsbits einer vorhandenen Datei oder eines Verzeichnisses in Oktalschreibweise zu ermitteln, verwenden Sie das Kommando stat.

### Optionen

- ▶ `-f` bzw. `--silent` bzw. `--quiet`  
zeigt keine Fehlermeldungen an.
- ▶ `-R` bzw. `--recursive`  
verändert auch die Zugriffsrechte von Dateien in allen Unterverzeichnissen.

### Beispiele

Die Datei `sichere` kann nun von allen Benutzern ausgeführt werden. `sichere` kann etwa ein Shell-Skript zur Erstellung eines Backups sein.

```
user$ chmod a+rx sichere
```

Das folgende Kommando entzieht allen Benutzern außerhalb der eigenen Gruppe die Lese- und Schreiberlaubnis für alle `*.odt`-Dateien im aktuellen Verzeichnis:

```
user$ chmod o-rw *.odt
```

Nur root darf die Datei `server.key` lesen, niemand darf sie verändern:

```
root# chmod 400 server.key
```

```
chown [optionen] user[:gruppe] dateien
```

`chown` ändert den Besitzer und (optional) auch die Gruppenzugehörigkeit einer Datei. Der Besitzer einer Datei kann nur von root verändert werden, während die Gruppe auch von anderen Benutzern eingestellt werden kann (siehe chgrp).

- ▶ `-R` bzw. `--recursive`  
verändert auch die Gruppenzuordnung von Dateien in allen Unterverzeichnissen.

### Beispiel

Die folgenden Kommandos stellen sicher, dass alle Dateien innerhalb von `/var/lib/mysql` dem Benutzer und der Gruppe `mysql` zugeordnet sind, sodass nur der MySQL-Server-Prozess die Dateien lesen und verändern darf:

```
root# chown -R mysql:mysql /var/lib/mysql
root# chmod -R o-rwx /var/lib/mysql
```

**chroot**verzeichnis [kommando]

Ohne weitere Parameter startet **chroot** eine neue Shell, die das angegebene Verzeichnis als Wurzelverzeichnis / verwendet. In dieser Shell können Sie interaktiv arbeiten. **exit** führt zurück in die ursprüngliche Shell.

Wenn Sie optional ein Kommando angeben, wird dieses Kommando statt der Shell gestartet. Während der Ausführung des Kommandos gilt abermals das angegebene Verzeichnis als Wurzelverzeichnis.

### Beispiel

**chroot** ist praktisch, um administrative Arbeiten in einem Dateisystem durchzuführen, das nicht beim Booten aktiviert wurde. Die folgenden Kommandos gehen davon aus, dass Sie einen Rechner von einem Rescue-System oder einer Live-DVD (Knoppix) gestartet haben. Nun binden Sie die Systempartition (im Beispiel `/dev/sda4`) in das Verzeichnis `/mnt/rescue` ein, machen dieses Verzeichnis zum neuen Wurzelverzeichnis und stellen dann mit **passwd** das vergessene **root**-Passwort neu ein:

```
root# mkdir /mnt/rescue
root# mount /dev/sda4 /mnt/rescue
root# chroot /mnt/rescue
root# passwd
Ändere Passwort für Benutzer root.
Geben Sie ein neues Passwort ein: *****
Geben Sie das neue Passwort erneut ein: *****
root# exit
root# reboot
```

**chsh** [user] shell

**chsh** verändert die Standard-Shell, die automatisch nach dem Einloggen aufgerufen wird. Zur Auswahl stehen alle in `/etc/shells` eingetragenen Shells, normalerweise `/bin/bash`, `/bin/csh` und `/bin/ksh`. Das Kommando **chsh** verändert die Datei `/etc/passwd` und trägt dort die neue Shell ein.

Jeder Anwender kann seine eigene Default-Shell nach Belieben verändern. Die Shell eines anderen Anwenders kann nur von **root** verändert werden. Die neue Shell muss mit dem vollständigen Verzeichnis angegeben werden. Die Änderung wird mit dem nächsten Login wirksam.

### chvt n

chvt aktiviert die Konsole n. Bei vielen Linux-Distributionen sind die Nummern 1 bis 6 für Textkonsolen reserviert. Auf der Konsole 7 läuft das Grafiksystem des ersten Benutzers. Falls sich mehrere Benutzer parallel im Display Manager einloggen, werden weitere Grafikkonsolen eingerichtet.

Bei Fedora läuft auf der Konsole 1 der Display-Manager im Grafikmodus. Konsole 2 ist für das Grafiksystem des ersten eingeloggten Benutzers vorgesehen. Die Konsolen 3 bis 6 werden im Textmodus betrieben.

### cksum datei

cksum ermittelt die Prüfsumme und die Länge der Datei in Bytes. Die Prüfsumme kann verwendet werden, um rasch festzustellen, ob zwei Dateien identisch sind. cksum liefert zuverlässigere Ergebnisse als das verwandte Kommando sum. Mathematisch noch sicherer sind md5sum oder sha512sum.

### clear

clear bzw. `[Strg]+[L]` löscht den Inhalt der Konsole.

### cmp [optionen] datei1 datei2

cmp vergleicht zwei Dateien Byte für Byte und liefert die Position der ersten Abweichung. Wenn die Dateien identisch sind, zeigt das Kommando überhaupt keine Meldung an. Zum Vergleich von Textdateien eignet sich diff in der Regel besser als cmp.

- ▶ `-c` bzw. `--show-chars`  
zeigt das jeweils erste Textzeichen an, bei dem sich die Dateien voneinander unterscheiden.
- ▶ `-l` bzw. `--verbose`  
liefert eine Liste aller Abweichungen.

**cnf** kommando

cnf steht für *Command Not Found* und ist ein SUSE-spezifisches Kommando, das ver-  
rät, in welchem noch nicht installierten Paket das gesuchte Kommando enthalten  
ist.

Unter Red-Hat- und Fedora-Distributionen enthält das von der bash ausgeführte  
Kommando `/usr/libexec/pk-command-not-found` eine vergleichbare Funktion, unter  
Ubuntu ist das `/usr/lib/command-not-found` aus dem Paket `command-not-found`. Dieses  
Paket steht auch unter Debian zur Verfügung. Dort muss es aber manuell installiert  
und durch das Kommando `update-command-not-found` einmalig eingerichtet werden.

**Beispiel**

```
user$ cnf jmacs
```

Das Programm 'jmacs' kann im folgenden Paket gefunden werden:

```
* joe [ Pfad: /usr/bin/jmacs, Repository: zypp (repo-oss) ]
```

Try installing with:

```
sudo zypper install joe
```

**compress** [optionen] datei

compress komprimiert bzw. dekomprimiert die angegebene Datei. Bei komprimierten  
Dateien wird die Kennung `.Z` an den Dateinamen angehängt.

compress existiert nur noch aus Kompatibilitätsgründen. Wesentlich leistungsfähiger  
sind bzip2, gzip und xz.

**continue** [n]

Das bash-Konstrukt `continue` überspringt in Shell-Scripts den Körper einer for-, while-  
oder until-Schleife und setzt die Schleife mit dem nächsten Durchlauf fort. Durch  
den optionalen Zahlenwert kann dieser Vorgang für mehrere Schleifenebenen durch-  
geführt werden.

**convert** [optionen] bildalt bildneu

convert aus dem Image-Magick-Paket konvertiert Bilddateien von einem Format in  
ein anderes. In der einfachsten Form wird es in der Art `convert name.tif name.jpg`  
aufgerufen, um die angegebene TIF-Datei in eine JPEG-Datei zu konvertieren. Die  
ursprüngliche Datei bleibt dabei erhalten. Mit über 100 Optionen können gleichzei-

tig diverse Bildparameter verändert werden. Die folgende Liste ist daher nur eine Auswahl.

- ▶ `-blur radius`  
verwischt das Bild.
- ▶ `-colors n`  
reduziert die Anzahl der RGB-Farben auf *n*.
- ▶ `-colorspace CMYK|GRAY|RGB|Transparent|YUV`  
gibt das gewünschte Farbmodell an (wobei zahlreiche weitere Modelle zur Auswahl stehen).
- ▶ `-compress None|BZip|Fax|Group4|JPEG|JPEG2000|Lossless|LZW|RLE|Zip`  
gibt das gewünschte Kompressionsformat an. Welche Formate tatsächlich zur Auswahl stehen, hängt allerdings vom Bildformat ab.
- ▶ `-contrast` bzw. `+contrast`  
verringert bzw. vergrößert den Kontrast des Bilds.
- ▶ `-crop geometry`  
schneidet den gewünschten Teil des Bilds aus. Beispielsweise beschreibt `-crop 50x50+100+100` ein 50 mal 50 Pixel großes Gebiet, das an der Koordinatenposition (100, 100) beginnt.
- ▶ `-filter Point|Box|Triangle|Hermite|Hanning|Hamming ...`  
wendet den gewünschten Filter auf das Bild an.
- ▶ `-gaussian radius`  
verwischt den Filter mit dem Gauß-Operator.
- ▶ `-normalize`  
normalisiert die Farbverteilung im Bild.
- ▶ `-quality n`  
gibt die gewünschte Kompressionsqualität an. Die zulässigen Werte für *n* hängen vom Bildformat ab (z. B. 0 bis 100 bei JPEG).
- ▶ `-resize NxN` bzw. `-resize n%`  
verändert die Auflösung des Bildes.



► `-rotate winkel`

dreht das Bild im Uhrzeigersinn um den angegebenen Winkel in Grad.

► `-trim`

schneidet einfarbige Bildränder ab, wenn sie dieselbe Farbe wie die Eckpunkte des Bilds haben.

## Beispiele

Die drei folgenden Beispiele zeigen einfache Anwendungen des Kommandos:

```
user$ convert -resize 100x100 in.jpg out.png
user$ convert -type Grayscale in.jpg out.eps
user$ convert -quality 80 in.bmp out.jpg
```

Das folgende Kommando macht aus einer PNG-Datei eine EPS-Datei gemäß der PostScript-Level-2-Spezifikation. (convert erzeugt normalerweise Dateien für PostScript Level 3. Diese sind dann oft deutlich größer, ohne dass sich daraus Vorteile ergeben.) Alle Bildebenen werden übereinander platziert. Transparente Bereiche der PNG-Datei werden weiß dargestellt, nicht schwarz.

```
user$ convert -flatten -alpha remove -background white in.png EPS2:out.eps
```

## Dokumentation

Die man-Seite zu convert ist nicht besonders hilfreich. Dafür sind sämtliche Optionen ausführlich auf der folgenden Webseite beschrieben:

<http://www.imagemagick.org/script/command-line-options.php>

**convmv** [optionen] dateien

Das Perl-Skript convmv von der Website <http://j3e.de/linux/convmv> bzw. aus dem Paket convmv ändert den Zeichensatz der angegebenen Dateinamen. Das Programm ist eine große Hilfe, wenn nach einer Zeichensatzumstellung Dateinamen falsch dargestellt werden.

► `-f zeichensatz`

gibt den ursprünglichen Zeichensatz der Dateinamen an. (convmv --list liefert eine Liste aller unterstützten Zeichensätze.)

► `-i`

ändert Dateinamen erst nach einer Rückfrage. Die Option ist nur in Kombination mit --notest zweckmäßig.

► `--notest`

ändert die Dateinamen tatsächlich. Ohne diese Option zeigt `convmv` lediglich die geplanten Änderungen an, ohne diese aber tatsächlich durchzuführen. Wenn Sie `convmv` rekursiv ausführen, ist ein Testlauf (also ohne `--notest`) unbedingt empfehlenswert!

► `-r`

wendet das Kommando rekursiv auf Unterverzeichnisse an.

► `-t zeichensatz`

gibt den neuen Zeichensatz an.

`convmv` verändert nur den Namen, nicht den Inhalt der Dateien! Wenn Sie den Inhalt von Textdateien von einem Zeichensatz in einen anderen ändern möchten, greifen Sie am besten auf die Kommandos `iconv` oder `recode` zurück.

### Beispiel

Um rekursiv alle Dateien eines Verzeichnisses vom Zeichensatz Latin-1 auf UTF-8 umzustellen (mit Rückfrage für jede einzelne Änderung), rufen Sie `convmv` so auf:

```
user$ convmv -r -i --notest -f iso-8859-1 -t utf8 verzeichnisname
```

```
cp [optionen] quelle ziel  
cp [optionen] dateien zielverzeichnis
```

`cp` kopiert Dateien und Verzeichnisse. Einzelne Dateien können beim Kopieren umbenannt werden. Bei der Bearbeitung mehrerer Dateien (z. B. durch die Angabe von Jokerzeichen) können diese lediglich in ein anderes Verzeichnis kopiert, nicht aber umbenannt werden. Anweisungen der Art `cp *.tex *.bak` sind nicht zulässig. Mit `cp` vergleichbare Kommandos sind `mv` zum Verschieben und Umbenennen von Dateien sowie `ln` zur Herstellung von Links. `cp` unterstützt unter anderem folgende Optionen:

► `-a` bzw. `--archive`

behält möglichst alle Attribute der Dateien bei. `-a` ist eine Abkürzung für `-dpR`.

► `-b` bzw. `--backup`

benennt bereits vorhandene Dateien in Backup-Dateien um (Dateiname plus `~`), anstatt sie zu überschreiben.

► `-d` bzw. `--dereference`

kopiert bei Links nur den Verweis, nicht aber die Datei, auf die der Link zeigt.

- ▶ `-i` bzw. `--interactive`  
fragt, bevor vorhandene Dateien überschrieben werden.
- ▶ `-l` bzw. `--link`  
erstellt feste Links (*Hard Links*), anstatt die Dateien zu kopieren. Wenn `cp` mit dieser Option verwendet wird, hat es dieselbe Funktionalität wie `ln`.
- ▶ `-p` bzw. `--preserve`  
lässt die Informationen über den Besitzer, die Gruppenzugehörigkeit, die Zugriffsrechte und den Zeitpunkt der letzten Änderung unverändert. Ohne diese Option gehört die Kopie demjenigen, der `cp` ausführt (Benutzer und Gruppe), und die Zeitangabe wird auf die aktuelle Zeit gesetzt.
- ▶ `-r` bzw. `-R` bzw. `--recursive`  
kopiert auch Unterverzeichnisse und die darin enthaltenen Dateien.
- ▶ `-s` bzw. `--symbolic-link`  
erstellt symbolische Links, anstatt die Dateien oder Verzeichnisse zu kopieren. `cp` hat damit die Funktionalität von `ln -s`.
- ▶ `-u` bzw. `--update`  
kopiert Dateien nur dann, wenn dabei keine gleichnamige Datei mit neuerem Datum überschrieben wird.

### Verzeichnisse kopieren

Wenn Sie ein ganzes Verzeichnis mit allen darin enthaltenen Dateien und Unterverzeichnissen kopieren möchten, führen Sie `cp -r quellverzeichnis zielverzeichnis` aus. Damit werden auch versteckte Dateien und Unterverzeichnisse kopiert. Wenn Sie möchten, dass beim Kopieren die Zugriffsrechte und -zeiten erhalten bleiben, verwenden Sie statt `-r` die Option `-a`.

Etwas diffizil ist die Frage, ob das Quellverzeichnis selbst oder nur sein Inhalt kopiert wird. Wenn es das Zielverzeichnis bereits gibt, wird darin das neue Unterverzeichnis `quellverzeichnis` erzeugt und der gesamte Inhalt des Quellverzeichnisses dorthin kopiert. Wenn es das Zielverzeichnis hingegen noch nicht gibt, wird es erzeugt; in diesem Fall wird nur der Inhalt des Quellverzeichnisses in das neu erzeugte Zielverzeichnis kopiert, nicht aber das Quellverzeichnis selbst. Zum Kopieren ganzer Verzeichnisbäume eignen sich auch `cpio`, `rsync` und `tar`.

## Beispiele

Das folgende Kommando kopiert alle \*.tex-Dateien aus dem Unterverzeichnis buch in das aktuelle Verzeichnis. Der Punkt gibt dabei als Zielverzeichnis das aktuelle Verzeichnis an:

```
user$ cp buch/*.tex .
```

Das zweite Kommando erstellt eine Backup-Kopie des gesamten Verzeichnisses buch:

```
user$ cp -a buch bak-buch
```

cp kann nicht dazu verwendet werden, mehrere Dateien beim Kopieren umzubenennen. cp \*.xxx \*.yyy kopiert also nicht alle \*.xxx-Dateien in \*.yyy-Dateien. Um solche Operationen durchzuführen, müssen Sie for oder sed zu Hilfe nehmen. Im folgenden Kommando bildet for eine Schleife über alle \*.xxx-Dateien. Der Ausdruck \${i%.xxx}.yyy entfernt die Endung \*.xxx und ersetzt sie durch .yyy. Wenn Sie cp durch mv ersetzen, werden die Dateien nicht kopiert, sondern umbenannt.

```
user$ for i in *.xxx; do cp $i ${i%.xxx}.yyy; done
```

Etwas komplizierter ist die Vorgehensweise mit sed: ls liefert die Liste der Dateien, die kopiert werden sollen, und gibt sie an sed weiter. sed bildet daraus mit dem Kommando s (*regular find and replace*) eine Liste von cp-Kommandos und gibt diese wiederum an eine neue Shell sh weiter, die die Kommandos schließlich ausführt.

```
user$ ls *.xxx | sed 's/\(.*\)\.xxx$/cp & \1.yyy/' | sh
```

### cpio kommando [optionen]

cpio fasst mehrere Dateien zu einem Archiv zusammen und kopiert sie auf einen anderen Datenträger, auf einen Streamer oder in ein anderes Verzeichnis. Analog kann das Kommando auch zum Wiedereinlesen solcher Daten verwendet werden. Unter Linux ist cpio eher ungebräuchlich, stattdessen wird zumeist tar verwendet. Die zentralen cpio-Kommandos sind:

#### ► -i

(*input*) liest ein Archiv aus der Standardeingabe und extrahiert die darin enthaltenen Dateien.

#### ► -o

(*output*) fasst Dateien in ein Archiv zusammen und schreibt diese an die Standardausgabe. Die zu verarbeitenden Dateien werden üblicherweise durch find ermittelt und dann über eine Pipe an cpio weitergegeben.

## ▶ -p

(*pass through*) überträgt Archive zwischen verschiedenen Verzeichnissen.

## ▶ -t

zeigt den Inhalt eines Archivs aus der Standardeingabe an.

Details dieser Aktionen können durch weitere Optionen gesteuert werden.

### Beispiel

Mit den folgenden Kommandos werden zuerst alle Dateien aus dem Verzeichnis `/etc` archiviert. `cpio -t` zeigt danach als Kontrolle die Liste der gesicherten Dateien.

```
root# cd /etc
root# find . | cpio -o > /tmp/etc-backup.cpio
root# cpio -t < /tmp/etc-backup.cpio
```

### **crontab** [optionen]

Das Kommando `crontab` hilft bei der Administration eigener Cron-Jobs in der Datei `/var/spool/cron/benutzername`. Cron-Jobs werden zu den vorgesehenen Zeiten automatisch im Hintergrund ausgeführt, z. B. einmal stündlich oder an jedem Sonntagabend.

## ▶ -e

öffnet einen Editor zur Bearbeitung der Crontab-Datei. Als Editor kommt normalerweise `vi` zum Einsatz. Wenn Sie einen anderen Editor wünschen, müssen Sie die Umgebungsvariable `EDITOR` entsprechend einstellen. `crontab` überprüft beim Speichern die Einhaltung der Crontab-Syntaxregeln.

## ▶ -l

listet die aktuellen Crontab-Einträge auf.

## ▶ -r

löscht alle Crontab-Einträge.

▶ -u *benutzername*

bearbeitet die Crontab-Datei des angegebenen Benutzers. Die Option `-u` steht nur `root` zur Verfügung.

```
cryptsetup [optionen] kommando
```

`cryptsetup` aus dem gleichnamigen Paket greift auf Funktionen des Kernelmoduls `dm_crypt` zurück. Es richtet Crypto-Devices ein, aktiviert und deaktiviert sie. Anstelle eines Devices kann auch eine ausreichend große Image-Datei verwendet werden. Diese kann vorweg z.B. mit `dd if=/dev/zero of=cryptoImage bs=1M count=n` erzeugt werden, wobei `n` die gewünschte Größe in MByte ist.

An dieser Stelle werden nur die wichtigsten LUKS-spezifischen Kommandos von `cryptsetup` beschrieben:

► `luksAddKey device`

richtet ein zusätzliches Passwort ein, das Zugriff auf den Crypto-Container gibt. Zur Ausführung des Kommandos muss ein bereits existierendes Passwort angegeben werden (egal welches). Insgesamt sind maximal acht Passwörter erlaubt.

► `luksClose device mappingname`

deaktiviert ein Crypto-Device.

► `luksDump device`

liefert Metainformationen über den Crypto-Container (z. B. den Verschlüsselungsalgorithmus).

► `luksFormat device`

richtet im angegebenen Device einen Crypto-Container ein. Dabei müssen Sie zweimal die *passphrase* angeben, die aus Sicherheitsgründen zumindest 20 Zeichen lang sein sollte. Standardmäßig wird zur Verschlüsselung der Algorithmus im Modus `aes-xts-plain64` mit einer Schlüssellänge von 256 Bit verwendet. Einen anderen Algorithmus können Sie mit `-c` angeben, eine andere Schlüssellänge mit `-s`. Welche Algorithmen der Kernel versteht, verrät die Pseudodatei `/proc/crypto`.

► `luksOpen device mappingname`

aktiviert den Crypto-Container im angegebenen Device und weist ihm einen Namen zu. Das resultierende Crypto-Device kann nun über `/dev/mapper/mappingname` genutzt werden.

## Beispiel

Die folgenden Kommandos zeigen, wie Sie einen USB-Stick (`/dev/sdh1`) zuerst als Crypto-Device formatieren (`luksFormat`) und das Device dann unter dem willkürlich gewählten Namen `mycontainer` aktivieren (`luksOpen`). Naturgemäß sind Ihre Daten nur

so sicher wie Ihr Passwort bzw. Ihre aus mehreren Wörtern bestehende Passphrase. Empfohlen wird eine Passwortlänge von zumindest 20 Zeichen.

Anschließend können Sie `/dev/mapper/mycontainer` wie eine Festplattenpartition oder ein Logical Volume nutzen – also ein Dateisystem einrichten, dieses in den Verzeichnisbaum einbinden etc. Nach `mount` müssen Sie daran denken, das Crypto-Device wieder zu deaktivieren (`luksClose`), um `/dev/sdh1` freizugeben. Erst jetzt dürfen Sie den USB-Stick ausstecken.

```
root# cryptsetup luksFormat /dev/sdh1
Daten auf /dev/sdh1 werden unwiderruflich überschrieben.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: *****
Verify passphrase: *****
Command successful.
root# cryptsetup luksOpen /dev/sdh1 mycontainer
Enter LUKS passphrase: *****
root# mkfs.ext4 /dev/mapper/mycontainer
root# mount /dev/mapper/mycontainer /test
root# ... das verschlüsselte Dateisystem nutzen ...
root# umount /test/
root# cryptsetup luksClose mycontainer
```

**csplit** [optionen] datei trennposition

`csplit` zerlegt eine Textdatei an vorgegebenen Stellen in mehrere Einzeldateien. Die Trennposition kann entweder durch eine direkte Zeilenangabe oder durch ein Suchmuster angegeben werden. Das Kommando liefert als Ergebnis die Dateien `xx00`, `xx01` etc. und gibt auf dem Bildschirm deren Längen aus. Durch die Angabe entsprechender Optionen sind natürlich auch »schönere« Dateinamen möglich.

`cat` kann aus diesen Einzeldateien wieder die Originaldatei zusammensetzen. Anstelle von `csplit` bietet sich zum Zerlegen binärer Dateien das Kommando `split` an.

### Angabe der Trennpositionen

Die Trennpositionen werden entweder durch eine Zeilenanzahl oder durch ein Suchmuster angegeben. Im einen Fall wird die Datei nach *n* Zeilen zerlegt, im anderen Fall vor oder nach dem Auftreten des Suchmusters. Wenn `csplit` die Datei mehrfach zerlegen soll, was zumeist der Fall ist, muss hinter der Zeilenanzahl bzw. dem Trennmuster angegeben werden, wie oft die Operation wiederholt werden soll.

- ▶ `/muster/`  
trennt die Datei in der Zeile vor dem Auftreten des Musters. (Die Zeile mit dem gefundenen Muster wird zur ersten Zeile der nächsten Datei.)
- ▶ `/muster/+n`  
`/muster/-n`  
trennt die Datei *n* Zeilen nach (+) oder vor (-) dem Auftreten des Musters.
- ▶ *n*  
trennt die Datei nach *n* Zeilen.
- ▶ `{n}`  
zerlegt die Datei in *n*+1 Einzeldateien (und nicht nur in zwei Dateien).
- ▶ `{*}`  
zerlegt die Datei bei jedem Auftreten des Suchmusters bzw. jedes Mal nach *n* Zeilen in die entsprechende Anzahl von Einzeldateien.

### Optionen

- ▶ `-f datei` bzw. `--prefix=datei`  
verwendet den angegebenen Dateinamen zur Benennung der Ausgabedateien.
- ▶ `-k` bzw. `--keep-files`  
Bereits erzeugte Dateien werden beim Auftreten eines Fehlers nicht wieder gelöscht. Die Option muss insbesondere bei Musterangaben in der Form *n* `{*}` verwendet werden. Die Musterangabe erfolgt wie bei grep.
- ▶ `-z` bzw. `--elide-empty-files`  
verhindert die Erzeugung leerer Dateien. Ohne diese Option können leere Dateien insbesondere dann auftreten, wenn bereits die erste Zeile der Ausgangsdatei dem Suchmuster entspricht.

### Beispiel

`csplit` zerlegt `total.txt` in die Dateien `teil.00`, `teil.01` etc. Die Einzeldateien sind jeweils 100 Zeilen lang. `cat` bildet daraus anschließend eine Kopie der Originaldatei.

```
user$ csplit -k -f teil. total.txt 100 {*}
user$ cat teil.* > kopie.txt
```



Im zweiten Beispiel zerlegt `csplit` die Datei `total.txt` in kleinere Dateien, wobei die Trennung immer dann erfolgt, wenn eine Zeile mit dem Text `% ===` beginnt.

```
user$ csplit -k -f teil. total.txt '/^% ===/' {*}
```

**curl** [optionen] [url]

`curl` hilft bei der Übertragung von Dateien von oder zu einem Server, wobei alle erdenklichen Protokolle unterstützt werden (HTTP, HTTPS, FTP, SFTP, SCP etc.). Die externe Datei bzw. das externe Verzeichnis wird durch eine URL-Zeichenkette (*Uniform Resource Locator*) angegeben, die mit dem Protokollnamen beginnt (z.B. `http://server.de/datei`).

► `--limit-rate n`

begrenzt die Übertragungsgeschwindigkeit auf die angegebene Byteanzahl pro Sekunde. `n` kann der Buchstabe `k` oder `m` hintangestellt werden, um die Übertragungsrate auf `n` kByte oder MByte pro Sekunde zu limitieren.

► `-o datei`

speichert die heruntergeladenen Daten in der angegebenen Datei, anstatt sie an die Standardausgabe weiterzuleiten.

► `-r n1-n2`

überträgt den angegebenen Bytebereich der Datei.

► `-T datei`

überträgt die angegebene Datei zum Server (*upload*). Statt des Dateinamens kann auch das Zeichen `-` angegeben werden, um Daten aus der Standardeingabe zu verarbeiten.

► `-u name:password`

gibt den Login-Namen und das Passwort an.

## Beispiele

Das folgende Kommando überträgt die angegebene Datei zum FTP-Server `backupserver` und speichert sie im Verzeichnis `verz`:

```
user$ curl -T datei -u username:password ftp://backupserver/verz
```

Um Daten aus dem Standardeingabekanal zu verarbeiten, geben Sie mit `-T` als Dateinamen einen Bindestrich an. Das folgende Kommando speichert das aus dem `tar`-Kommando resultierende Ergebnis direkt in der Datei `name.tgz` auf dem FTP-Server:

```
user$ tar czf - verzf/ | curl -T - -u user:pw ftp://bserver/name.tgz
```

**cut** [optionen] datei

`cut` extrahiert aus jeder Zeile eines Textes die durch Optionen angegebenen Spalten.

► `-b liste` bzw. `--bytes liste`

extrahiert die in einer Liste angegebenen Zeichen. Einzelne Einträge dürfen durch Kommata (aber nicht durch Leerzeichen) getrennt werden. Statt einzelner Zeichen dürfen auch ganze Bereiche angegeben werden, etwa `-b 3-6,9,11-15`.

► `-d zeichen` bzw. `--delimiter zeichen`

gibt das Trennzeichen für `-f` an, das statt des Tabulatorzeichens verwendet werden soll.

► `-f liste` bzw. `--fields liste`

wie `-b`, aber jetzt für Felder (Datensätze), die durch Tabulatorzeichen getrennt sein müssen.

► `-s` bzw. `--only-delimited`

eliminiert alle Zeilen, die keine Daten enthalten, die der Option `-f` entsprechen. Kann nicht zusammen mit `-b` verwendet werden.

In der Praxis gelingt die Extraktion einzelner Textspalten mit `awk` oft wesentlich unkomplizierter als mit `cut`.

### Beispiel

Im folgenden Beispiel liest `cut` aus der Datei `/etc/passwd` die erste, die dritte und die siebte Spalte, also den Benutzernamen, die UID-Nummer und die Default-Shell. Das Trennzeichen in dieser Datei ist der Doppelpunkt.

```
root# cut -d\: -f1,3,7 /etc/passwd
root:0:/bin/bash
bin:1:/sbin/nologin
daemon:2:/sbin/nologin
adm:3:/sbin/nologin
...
```

```
date [optionen] [+format]
```

`date` liefert das aktuelle Datum und die Uhrzeit bzw. verändert diese Daten. Eine Veränderung darf nur von `root` vorgenommen werden. `date` kann die Zeit in den verschiedensten Formaten anzeigen.

► `-s neuezeit`

verändert das Datum und/oder die Uhrzeit.

Die folgende Aufzählung nennt die wichtigsten Formatcodes. Die Formatzeichenkette muss in Anführungszeichen gesetzt werden.

► `%Y`

das Jahr als vierstellige Zahl.

► `%l`

der Monat als Zahl (1 bis 12).

► `%m`

der Monat als zweistellige Zahl (01 bis 12).

► `%b`

der Monat als kurze Zeichenkette (Jan bis Dez).

► `%B`

der Monat als lange Zeichenkette (Januar bis Dezember).

► `%d`

der Tag des Monats als zweistellige Zahl (01 bis 31).

► `%e`

der Tag des Monats als Zahl (1 bis 31), wobei einstelligen Zahlen ein Leerzeichen vorangestellt wird.

► `%H`

die Stunde als zweistellige Zahl (00 bis 23).

► `%M`

die Minute als zweistellige Zahl (00 bis 59).

### Beispiel

Das folgende Script erzeugt ein komprimiertes tar-Archiv des Verzeichnisses `/home/kofler/data`. Das Archiv wird unter zwei Dateinamen gespeichert: `mydata-day-dd.tar.gz` und `mydata-month-mm.tar.gz`. Wenn das Script täglich ausgeführt wird, haben Sie mit der Zeit 43 Backup-Versionen, die den Zustand des Backup-Verzeichnisses für die letzten 28 bis 31 Tage sowie für die letzten 12 Monate widerspiegeln.

```
#!/bin/bash
fname1=/backup/mydata-day-$(date "+%d").tar.gz
fname2=/backup/mydata-month-$(date "+%m").tar.gz
tar czf $fname1 /home/kofler/data
cp $fname1 $fname2
chmod 600 $fname1 $fname2
```

### dcfldd optionen

`dcfldd` aus dem gleichnamigen Paket ist eine Alternative zum Low-Level-Kopierkommando `dd`. Im Unterschied zu `dd` kann `dcfldd` ein optisches Feedback während der Ausführung zeitaufwendiger Kopierkommandos geben. Eine wirklich präzise Fortschrittsanzeige kann aber auch `dcfldd` nicht bieten, weil die Ergebnisse durch das I/O-Caching verfälscht werden: Zuerst scheint alles ganz schnell zu gehen, dann dauert es aber doch recht lange, bis der Vorgang abgeschlossen ist.

`dcfldd` wird durch dieselben Optionen wie `dd` gesteuert. Darüber hinaus gibt es ein paar zusätzliche Optionen, deren wichtigste `statusinterval` lautet:

#### ► `statusinterval=n`

gibt an, nach wie vielen Blöcken jeweils der Kopierfortschritt angezeigt werden soll. Standardmäßig beträgt *n* 256.

### Beispiel

Das folgende Kommando kopiert die Image-Datei der Raspberry-Pi-Distribution Raspian auf eine SD-Karte. Beachten Sie, dass der Device-Name zum Zugriff auf die SD-Karte je nach Anzahl der internen Festplatten oder SSDs variiert. Passen Sie auf, dass Sie nicht irrtümlich einen internen Datenträger überschreiben! Der Vorgang kann nicht rückgängig gemacht werden.

```
root# dcfldd if=raspbian.img of=/dev/sdb bs=4M statusinterval=10
```

## dd optionen

dd überträgt Daten zwischen verschiedenen Speichermedien (Festplatte, USB-Stick etc.) und führt dabei auf Wunsch eine Konvertierung der Daten durch. Das Kommando kann dazu eingesetzt werden, Daten zwischen verschiedenen Rechnerarchitekturen auszutauschen.

dd kann nicht nur einzelne Dateien kopieren, sondern auch direkt auf Devices zugreifen. Damit können ganze Festplatten(partitionen) kopiert werden und kann der Bootsektor der Festplatte verändert werden etc. Auf dem Datenträger muss kein Dateisystem eingerichtet werden. Wenn dd ohne Optionen verwendet wird, liest es die Daten aus der Standardeingabe (Ende mit `[Strg]+[Z]`) und schreibt in die Standardausgabe. Die Optionen von dd werden ohne vorangehende Minus-Zeichen angegeben!

► `bs=n`

bestimmt die Blockgröße für die Ein- und Ausgabedatei. (Die Blockgröße gibt an, wie viele Bytes jeweils in einem Durchgang gelesen bzw. geschrieben werden.) Vorsicht: Standardmäßig verwendet dd eine Blockgröße von nur 512 Byte. Das ist extrem ineffizient! Wenn Sie mit dd Festplatten oder SSDs auslesen bzw. beschreiben, sollten Sie zumindest mit `bs=1M` arbeiten. Das Suffix `M` steht für  $1024 \times 1024$  Byte, `K` für 1024 Byte.

► `conv=modus`

konvertiert die Daten während des Kopierens. Für *modus* sind verschiedene Einstellungen erlaubt, unter anderem `lcase` (Großbuchstaben in Kleinbuchstaben umwandeln), `ucase` (Klein- in Großbuchstaben umwandeln), `swab` (jeweils zwei Byte vertauschen) etc.

► `count=n`

kopiert nur *n* Blöcke (und nicht die gesamten Daten).

► `ibs=n`

bestimmt die Blockgröße der Quelldatei.

► `if=quelldatei`

gibt die Quelldatei (statt der Standardeingabe) an.

► `obs=n`

bestimmt die Blockgröße der Zieldatei.

► `of=zieldatei`

gibt die Zieldatei (statt der Standardausgabe) an.

- ▶ `seek=n`  
überspringt  $n$  Blöcke, bevor die Ausgabe gestartet wird.
- ▶ `skip=n`  
überspringt  $n$  Blöcke, bevor mit dem Lesen begonnen wird.

Beachten Sie, dass der direkte Zugriff auf Devices (z. B. in der Form `cat /dev/xxx >`) oft schneller ist als `dd`. Außerdem müssen Sie sich dann keine Gedanken über die optimale Blockgröße machen.

### Beispiele

Durch das folgende Kommando wird die ISO-Datei `opensuse.iso` direkt auf den Datenträger mit dem Device `/dev/sdc` übertragen, also beispielsweise auf einen USB-Stick. Beachten Sie, dass dabei alle bisher gespeicherten Daten auf diesem Datenträger verloren gehen! Stellen Sie deswegen sicher, dass Sie den richtigen Device-Namen angeben. Wenn das mit `of` angegebene Device Ihre aktuelle Festplatte ist, verlieren Sie deren Inhalt!

```
root# dd if=opensuse.iso of=/dev/sdc bs=1M
```

Das zweite Kommando erzeugt eine 10 GByte große RAW-Image-Datei für ein Virtualisierungsprogramm (z. B. QEMU/KVM). Diese Image-Datei besteht einfach aus Nullen, die aus dem Device `/dev/zero` gelesen werden.

```
root# dd if=/dev/zero of=image.raw bs=1M count=10000
```

```
declare [optionen] var[=wert]
```

Das `bash`-Kommando `declare` weist Shell-Variablen einen neuen Wert und/oder diverse Eigenschaften zu. Das Kommando wird vor allem in Shell-Scripts verwendet. Wenn es ohne Parameter aufgerufen wird, werden alle bekannten Variablen mit ihrem Inhalt aufgelistet. Vor und nach dem Gleichheitszeichen dürfen keine Leerzeichen angegeben werden!

- ▶ `-A`  
deklariert die Variable als assoziatives Array.
- ▶ `-r`  
deklariert die Variable `read-only`. Die Variable darf nur noch gelesen, aber nicht mehr verändert werden.

## ► -x

deklariert die Variable als Umgebungsvariable (wie durch `export`). Die Variable steht damit auch anderen Kommandos bzw. in Subshells zur Verfügung.

**delgroup** [optionen] name

`delgroup` löscht auf Debian- und Ubuntu-Systemen die angegebene Gruppe und befolgt dabei die Regeln aus /etc/deluser.conf. Bei Fedora und Red Hat ist `delgroup` ein Link auf das Kommando groupdel, das aber andere Optionen kennt.

## ► --only-if-empty

löscht die Gruppe nur, wenn ihr keine Benutzer zugeordnet sind.

## ► --system

löscht die Gruppe nur, wenn es sich um eine Systemgruppe handelt.

**deluser** [optionen] name

`deluser` löscht auf Debian- und Ubuntu-Systemen den angegebenen Benutzer und befolgt dabei die Regeln aus /etc/deluser.conf. Demzufolge wird das Heimatverzeichnis normalerweise nicht gelöscht. Bei Fedora und Red Hat ist `deluser` ein Link auf das Kommando userdel mit anderen Optionen.

## ► --remove-all-files

löscht das Heimatverzeichnis sowie alle anderen Dateien des Benutzers (z. B. diverse Spool- und Mail-Dateien).

## ► --remove-home

löscht auch das Heimatverzeichnis des Benutzers.

## ► --system

löscht den Benutzer nur, wenn es sich um einen Systembenutzer handelt.

**depmod** [optionen]

`depmod` erzeugt die Modulabhängigkeitsdatei `modules.dep` sowie diverse `*.map`-Dateien im Verzeichnis `/lib/modules/kernelversion/`. Die `*-map`-Dateien geben an, bei welcher Hardware-Komponente welches Kernelmodul geladen werden soll.

- ▶ `-A` bzw. `--quick`  
testet zuerst, ob bereits existierende \*.dep- und \*.map-Dateien noch aktuell sind. In diesem Fall verzichtet depmod darauf, die Dateien neu zu generieren.
- ▶ `-b verzeichnis` bzw. `--basedir verzeichnis`  
gibt an, für welches Verzeichnis die \*.dep- und \*.map-Dateien erzeugt werden sollen. Standardmäßig aktualisiert das Kommando die Dateien für den laufenden Kernel.

**df** [optionen] [verzeichnis]

df gibt Auskunft darüber, an welcher Stelle im Dateibaum Festplatten(partitionen) oder andere Laufwerke eingebunden sind und wie viel Speicherplatz darauf verfügbar ist. Normalerweise liefert df eine Liste aller aktiven Partitionen. Wenn Sie an das Kommando ein Verzeichnis übergeben, zeigt es hingegen nur die Daten der Partition an, in der sich dieses Verzeichnis physikalisch befindet. Bei Systemen mit vielen Partitionen können Sie auf diese Weise einfach feststellen, in welcher Partition ein bestimmtes Verzeichnis gespeichert wird.

- ▶ `-h`  
gibt den Speicherplatz in leicht lesbarer Form in MByte oder GByte an.
- ▶ `-i` bzw. `--inodes`  
gibt Informationen über die verfügbaren I-Nodes an (statt des freien Speicherplatzes in kByte).
- ▶ `-T`  
gibt für jede Partition bzw. für jeden Datenträger das Dateisystem an.

### Beispiel

Auf dem Beispielsystem sind in der Systempartition / noch 10 GByte Speicherplatz frei, in der Bootpartition noch ca. 340 MByte. Die anderen Dateisysteme dienen nur Verwaltungszwecken und sind als temporäre Dateisysteme ausgeführt (tmpfs), deren Inhalt beim Herunterfahren des Rechners verloren geht.

```
root# df -h
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/mapper/fedora-root 16G   4,6G  10G  32% /
/dev/sda1        477M   108M  341M  24% /boot
devtmpfs         622M     0  622M   0% /dev
```



tmpfs	632M	364K	632M	1% /dev/shm
tmpfs	632M	1,2M	631M	1% /run
tmpfs	632M	0	632M	0% /sys/fs/cgroup
tmpfs	632M	24K	632M	1% /tmp
tmpfs	127M	16K	127M	1% /run/user/42
tmpfs	127M	32K	127M	1% /run/user/1000

### dhclient schnittstelle

Das Kommando `dhclient` aus dem `dhcp-client`-Paket bezieht bei den meisten Distributionen die Parameter für die Netzwerkkonfiguration von einem DHCP-Server. Das Kommando eignet sich ideal für eine Ad-hoc-Netzwerkkonfiguration bei neuen Server-Installationen. Bei manchen Distributionen kommen als Alternativen zu `dhclient` die Kommandos `dhclient3` oder `dhcpcd` zum Einsatz.

### Beispiel

Das folgende Kommando führt eine automatische Netzwerkkonfiguration für die Schnittstelle `eth0` durch:

```
root# dhclient eth0
```

### diff [optionen] datei1 datei2

`diff` vergleicht zwei Textdateien. Das Ergebnis ist eine Liste aller Zeilen, die voneinander abweichen. Das Kommando ist relativ »intelligent«, d. h., wenn in einer Datei gegenüber der anderen einige Zeilen eingefügt sind, so wird nur diese Abweichung gemeldet. Weitere Zeilen werden wieder als identisch erkannt, obwohl jetzt unterschiedliche Zeilennummern vorliegen. Das Kommando kann also dazu verwendet werden, die Abweichungen zwischen zwei Versionen eines Programmlistings rasch zu dokumentieren.

- ▶ `-b`  
betrachtet mehrfache Leerzeichen und -zeilen wie einfache Leerzeichen bzw. -zeilen.
- ▶ `-w`  
ignoriert Leerzeichen und Leerzeilen ganz.

### Beispiel

Das folgende Kommando vergleicht den Inhalt der eigenen MySQL-Konfigurationsdatei mit der vom Paket vorgeschlagenen Konfiguration. Die mit < beginnenden Zeilen zeigen den aktuellen Zustand an (also Text aus der ersten an diff übergebenen Datei), die mit > beginnenden Zeilen zeigen die entsprechende Passage aus der zweiten Datei.

```
root# diff /etc/mysql/my.cnf /etc/mysql/my.cnf.dpkg-dist
100,101c87,88
< server-id                = 100
< log_bin                  = /var/log/mysql/mysql-bin.log
---
> #server-id               = 1
> #log_bin                 = /var/log/mysql/mysql-bin.log
...
```

**dircolors** [optionen] [datei]

dircolors liefert den Programmcode zur Einstellung der Farben, die das Kommando ls zur Kennzeichnung verschiedener Dateitypen verwendet. Falls an dircolors eine Datei übergeben wird, zeigt das Kommando die in dieser Datei gespeicherten Einstellungen. Die Datei muss die durch dircolors -p vorgegebene Syntax einhalten.

- ▶ -b  
verwendet die Syntax der Shell bash (gilt standardmäßig).
- ▶ -c  
verwendet die Syntax der C-Shell.
- ▶ -p  
verwendet eine sehr ausführliche Syntax, die von dircolors selbst verarbeitet werden kann. Die Datei enthält Kommentare, die die Syntax und insbesondere die Farbcodes erläutern.

### Beispiel

Bei vielen Distributionen wird beim Start der bash die Datei .dircolors ausgewertet und zur Einstellung der Farben berücksichtigt. Um die ls-Farben zu ändern, erzeugen Sie zuerst mit dircolors -p eine eigene Version dieser Datei und verändern diese dann mit einem Editor. Die geänderten Einstellungen werden wirksam, sobald Sie ein neues Terminal-Fenster oder Dialogblatt öffnen.

```
user$ cd (in das Heimatverzeichnis wechseln)
user$ dircolors -p > .dircolors
user$ ihrLieblingsEditor .dircolors
```

Sollte die Veränderung der Datei `.dircolors` bei Ihrer Distribution nicht ausreichen, erzeugen Sie daraus mit `dircolor` eine Anweisung zur Einstellung der Variable `LS_COLORS` und fügen diese Anweisung am Ende der Datei `.bashrc` ein:

```
user$ dircolors .dircolors >> .bashrc
```

### **dirname** zeichenkette

`dirname` liefert den Pfad eines vollständigen Dateinamens. `dirname /usr/bin/groff` liefert also `/usr/bin`.

### **dirs**

Das `bash`-Kommando `dirs` liefert die Liste aller durch `pushd` gespeicherten Verzeichnisse. Das Kommando ist primär zur Script-Programmierung gedacht.

### **disown** [optionen] [jobspec]

Das `bash`-Kommando `disown` löst den angegebenen Prozess von der Shell. Damit läuft der Prozess weiter, selbst wenn die Shell beendet wird. Der Prozess wird wahlweise durch die Prozess-ID-Nummer (PID) oder durch die Shell-Jobnummer angegeben, wobei der Shell-Jobnummer das Zeichen `%` vorangestellt werden muss. Ohne die Angabe einer Jobnummer ist der aktuelle (Hintergrund-)Prozess gemeint.

Wenn Sie von vornherein planen, ein Programm losgelöst von der Shell auszuführen, können Sie bereits beim Start auf `nohup` zurückgreifen und sparen sich so den späteren Aufruf von `disown`.

#### ► -a

löst alle in der Shell gestarteten Prozesse.

#### ► -r

löst alle in der Shell gestarteten Prozesse, die zurzeit ausgeführt werden.

### Beispiel

Im folgenden Beispiel wird zuerst ein längerer Download-Prozess gestartet. `[Strg]+[Z]` unterbricht den Download, `bg` setzt ihn im Hintergrund fort. `disown` löst den

Download-Prozess von der Shell, sodass der Download auch nach dem Shell-Logout fortgesetzt wird.

```
user$ wget -q http://mirror.switch.ch/.../CentOS-n.n.iso
user$ <Strg>+<Z>
user$ bg
user$ disown
```

### **dmesg** [optionen]

**dmesg** gibt die im Pufferspeicher enthaltenen Kernelmeldungen aus.

#### ► -C

löscht nach der Ausgabe den Pufferspeicher für die Kernelmeldungen. Diese Option erfordert `root`-Rechte.

### **dnf** [optionen] kommando

**dnf** ersetzt seit Fedora 21 das Paketverwaltungskommando `yum`. RHEL 7 und CentOS 7 sind vorerst bei `yum` geblieben, dort ist die Umstellung auf **dnf** in Version 8 zu erwarten.

Obwohl **dnf** vollkommen neu implementiert wurde, sind die Grundkommandos syntaxkompatibel zu `yum`. Die Beschreibung der wichtigsten Paketverwaltungsbefehle und Optionen finden Sie deswegen beim Kommando `yum`.

Es gibt natürlich Unterschiede zwischen `yum` und **dnf**. Diese betreffen aber eher selten benötigte Kommandos und Optionen. Details können Sie hier nachlesen:

[http://dnf.readthedocs.org/en/latest/cli\\_vs\\_yum.html](http://dnf.readthedocs.org/en/latest/cli_vs_yum.html)

### **dpkg** optionen [dateiname/paketname]

**dpkg** erledigt die Low-Level-Paketverwaltung in allen Debian- und Ubuntu-basierten Distributionen. Der Benutzer verwendet aber in der Regel nicht **dpkg** zur Paketinstallation, sondern ein darauf aufbauendes Kommando, meist `apt-get`.

#### ► --configure paketname

führt die Konfigurations-Skripts des angegebenen Pakets aus. Normalerweise geschieht das bereits während der Installation; in manchen Fällen kann es aber notwendig sein, diesen Schritt explizit auszuführen.

Bei manchen Paketen gibt es zusätzlich zu den automatischen Konfigurations-Scripts interaktive Setup-Programme. Wenn Sie ein derartiges Setup-Programm später nochmals benötigen, müssen Sie `dpkg-reconfigure paketname` ausführen.

► `--get-selections`

liefert ähnlich wie `--list` eine Liste aller installierten Pakete. Das Ergebnis enthält aber weniger Detailinformationen und ist damit viel übersichtlicher. Wenn Sie die Liste in eine Textdatei umleiten und speichern, können Sie alle Pakete später auf einem anderen Rechner mit `--set-selections` installieren.

► `--install dateiname.deb`

installiert die angegebene(n) Paketdatei(en). Wenn bereits eine ältere Version installiert ist, wird diese deinstalliert und durch die neue Version ersetzt. Während der Installation werden auch die im Paket vorgesehenen Konfigurations-Scripts ausgeführt. Vor der Installation wird sichergestellt, dass alle Paketabhängigkeiten erfüllt sind. Wenn das nicht der Fall ist, geht aus der Fehlermeldung zumeist hervor, welche Pakete fehlen.

`dpkg` lässt normalerweise nur die Installation von Paketen in der zur Distribution passenden Architektur zu. Wenn Sie mit einer 64-Bit-Linux-Distribution arbeiten, ein Programm aber nur als 32-Bit-Paket verfügbar ist, können Sie die Installation mit `--force-architecture` erzwingen. Ob das Programm dann tatsächlich funktioniert, ist eine andere Frage. Auf jeden Fall müssen Sie auch alle erforderlichen 32-Bit-Bibliotheken installieren.

► `--list`

liefert eine Liste aller installierten Pakete.

► `--list 'muster'`

liefert eine Liste der installierten und verfügbaren Pakete. Bei der Paketliste wird in der ersten Spalte ein Code aus zwei Buchstaben angegeben. Der erste Buchstabe gibt den gewünschten Status des Pakets an (`i` = installieren, `n` = nicht installieren, `r/p` = entfernen, `h` = halten), der zweite Buchstabe den tatsächlichen Status (`i` = installiert, `n` = nicht installiert, `c` = konfiguriert, `u` = entpackt, aber noch nicht konfiguriert, `f` = fehlgeschlagen).

Die Informationen stammen aus der Debian-Paketdatenbank, einer Sammlung von Dateien im Verzeichnis `/var/lib/dpkg`. Dort werden Meta-Informationen über alle installierten und verfügbaren Pakete gespeichert.

► `--listfiles paketname`

liefert eine Liste aller Dateien des angegebenen Pakets. Das funktioniert nur für bereits installierte Pakete. Den Inhalt nicht installierter Pakete ermitteln Sie mit dem Kommando `dpkg-deb --contents dateiname`.

► `--purge paketname`

entfernt das angegebene Paket inklusive aller Paketdateien (auch wenn diese von Ihnen verändert wurden).

► `--remove paketname`

entfernt das angegebene Paket.

► `--search dateiname`

ermittelt das Paket, aus dem die angegebene Datei stammt.

### Beispiel

Das erste `dpkg`-Kommando installiert ein neues Paket. `dpkg --search` ermittelt das Paket, von dem die Datei `/etc/sensors3.conf` stammt. `dpkg --listfiles` liefert eine Liste aller Dateien dieses Pakets.

```
root# dpkg --install paketname.deb
root# dpkg --search /etc/sensors3.conf
libsensors4: /etc/sensors3.conf
root# dpkg --listfiles libsensors4
...
/usr/lib
/usr/lib/libsensors.so.4.2.1
/etc
/etc/sensors3.conf
/etc/sensors.d
...
```

**dracut** [optionen] initrd-datei kernelversion

`dracut` ist unter CentOS, Fedora, RHEL und (open)SUSE für das Erzeugen einer Initrd-Datei zuständig. Wenn `dracut` ohne weitere Parameter ausgeführt wird, erzeugt es für den neuesten Kernel im Verzeichnis `/boot` eine Initrd-Datei mit dem Namen `/boot/initrd-kernelversion`. `dracut` berücksichtigt die Einstellungen in `/etc/dracut.conf` sowie zahllose Optionen, die in `man dracut` beschrieben sind. Hier sind nur die drei wichtigsten Optionen zusammengefasst:

- ▶ `-d a,b,c`  
baut die Kernelmodule `a`, `b` und `c` in die Initrd-Datei ein. Normalerweise ist diese Option nicht erforderlich. `dracut` erkennt selbstständig, welche Module für den Startprozess erforderlich sind.
- ▶ `-f`  
überschreibt eine vorhandene Initrd-Datei.
- ▶ `--regenerate-all`  
erzeugt neue Initrd-Dateien zu allen auf dem System installierten Kernel-Versionen.

Bei älteren CentOS-, Fedora-, RHEL- und SUSE-Versionen kommt anstelle von `dracut` das Kommando `mkinitrd` zum Einsatz. Debian und Ubuntu verwenden stattdessen `update-initramfs`.

## Beispiel

Um für einen selbst kompilierten Kernel 4.3.3 in der Datei `/boot/vmlinuz-4.3.3` manuell eine Initrd-Datei zu erzeugen, führen Sie das folgende Kommando aus:

```
root# dracut /boot/initrd-4.3.3 4.3.3
```

**du** [optionen] [verzeichnis]

`du` gibt Informationen über den Speicherbedarf von Dateien bzw. von Verzeichnissen aus. Wenn im Verzeichnisparameter eine Dateispezifikation angegeben wird (etwa `*` oder `*.tex`), dann liefert `du` eine Liste mit der Größe aller Dateien. Wird dagegen nur ein Verzeichnis angegeben, ermittelt `du` den Speicherbedarf für alle untergeordneten Verzeichnisse. Die Speicherangaben umfassen auch den Speicherbedarf aller untergeordneten Verzeichnisse. Der letzte Zahlenwert gibt den Gesamtspeicherbedarf aller Dateien und Unterverzeichnisse ab dem angegebenen Verzeichnis an. Alle Angaben erfolgen in kByte. `du` bietet keine Möglichkeit, das Ergebnis zu sortieren.

- ▶ `-b` bzw. `--bytes`  
zeigt die Größenangaben in Byte (statt in kByte) an.
- ▶ `-c` bzw. `--total`  
zeigt als abschließenden Wert die Endsumme an. Diese Option ist nur notwendig, wenn `du` auf Dateien (und nicht auf Verzeichnisse) angewandt wird. Mit dieser Option kann relativ einfach festgestellt werden, wie viel Speicher alle Dateien mit einer bestimmten Kennung (z. B. `*.pdf`) beanspruchen.

- ▶ `-h` bzw. `--human-readable`  
zeigt die Größenangaben in einer gut lesbaren Form an. K, M und G gelten als Abkürzungen für kByte, MByte oder GByte.
- ▶ `--max-depth=n` bzw. `--max n`  
gibt die Verzeichnisgröße nur für die angegebene Anzahl von Verzeichnisebenen aus.
- ▶ `-s` bzw. `--summarize`  
zeigt *nur* die Endsumme an. Diese Option ist nur dann sinnvoll, wenn der Speicherbedarf von Verzeichnissen angezeigt wird.
- ▶ `-S` bzw. `--dereference`  
zeigt nur den Speicherbedarf unmittelbar im Verzeichnis an. Der Speicherbedarf in Unterverzeichnissen wird *nicht* berücksichtigt.

### Beispiel

du ermittelt, welche Verzeichnisse wie viel Platz beanspruchen. Mit `--max 1` werden Unterverzeichnisse nicht einzeln angeführt, sind aber im Ergebnis berücksichtigt.

```
user$ du -h --max 1
15M   ./Downloads
29M   ./Videos
69M   ./mozilla
...
```

```
dumpe2fs device
```

`dumpe2fs` liefert unzählige interne Informationen über den Zustand eines ext-Dateisystems, insbesondere über die sogenannten Superblöcke sowie über die Organisation der Blockgruppen des Datenträgers.

- ▶ `-b`  
listet nur defekte Blöcke des Datenträgers auf.
- ▶ `-h`  
liefert nur eine Zusammenfassung der Daten des Superblocks.



```
dvd+rw-format [optionen] device
```

dvd+rw-format formatiert DVD+RW-, DVD-RW- und DVD-RAM-Medien.

► -blank

formatiert DVD-RW-Medien im Modus *incremental sequential* (anstatt standardmäßig im Modus *restricted overwrite*). Das hat den Vorteil, dass eine höhere Kompatibilität für die Verwendung als Video-DVD erreicht wird. Allerdings dauert das Formatieren sehr lange; außerdem kann `growisofs` vorhandene Daten nicht überschreiben. Die Option ist ausschließlich für DVD-RWs vorgesehen.

► -force

erzwingt die Formatierung einer bereits formatierten DVD-RW oder DVD+RW.

► -format=full

erzwingt die vollständige Formatierung einer bereits formatierten DVD-RAM.

► -lead-out

schreibt einen *Lead-out*-Bereich auf eine bereits beschriebene DVD+RW. Diese Option ist nur für DVD+RWs gedacht und erhöht die Kompatibilität zu manchen DVD-Lesegeräten. Wenn Sie diese Option verwenden, wird die DVD nicht formatiert!

```
dvd+rw-mediainfo device
```

dvd+rw-mediainfo liefert Informationen über den im Laufwerk enthaltenen Datenträger (DVD-Typ, Formatierung, Session-Informationen etc.).

```
dvips [optionen] name.dvi
```

dvips erzeugt aus einer \*.dvi-Datei eine PostScript-Datei. Wenn dvips ohne die Option -o eingesetzt wird, leitet das Kommando die resultierende PostScript-Datei an den Standarddrucker weiter.

► -A

wandelt nur ungerade Seiten um.

► -B

wandelt nur gerade Seiten um.

► `-D n`

verwendet bei der Erzeugung von  $\text{\LaTeX}$ -Bitmap-Schriften eine Auflösung von *n* dpi (dots per inch). Die Standardauflösung beträgt meist 600 dpi (siehe `/etc/texmf/config.ps`). Alternativ darf *n* auch 300, 400 oder 1270 betragen (siehe `/usr/bin/mktexpk`). Die Option ist nur für Bitmap-Schriften relevant. PostScript-Schriften sind immer auflösungsunabhängig.

► `-E`

erzeugt eine EPS-Datei (Encapsulated PostScript) mit einer Boundingbox, die nur den tatsächlich genutzten Teil der Seite umfasst. Das ist nur sinnvoll, wenn die DVI-Datei nur eine Seite umfasst und die resultierende EPS-Datei anschließend in ein anderes Dokument eingebettet werden soll.

► `-i -S n`

zerlegt die Ausgabe in Dateien zu je *n* Seiten. Die Dateien werden automatisch durchnummeriert.

► `-l letzteseite`

beendet die Umwandlung mit der angegebenen Seite.

► `-o zieldatei`

schreibt das Ergebnis in die angegebene Datei (anstatt es an das Programm `lpr` weiterzuleiten).

► `-p ersteseite`

beginnt die Umwandlung mit der angegebenen Seite.

► `-pp n1,n2-n3,n4,n5,n6-n7`

druckt die angegebenen Seiten. Beachten Sie, dass in der Seitenliste keine Leerzeichen vorkommen dürfen.

Globale Standardeinstellungen für `dvips` sind in der Datei `config.ps` definiert. Diese Datei befindet sich je nach Distribution z. B. im Verzeichnis `/etc/texmf` oder in `/usr/share/texlive/texmf-dist/dvips/config`.

**e4defrag** [*optionen*] *datei/verzeichnis/device*

`e4defrag` defragmentiert wahlweise eine Datei, den gesamten Inhalt eines Verzeichnisses oder ein ganzes `ext4`-Dateisystem. Benutzer ohne `root`-Rechte können nur ihre eigenen Dateien defragmentieren.

Defragmentierung bedeutet, dass die Blöcke einer Datei möglichst aneinanderreihend auf dem Datenträger reserviert werden. Das beschleunigt besonders bei herkömmlichen Festplatten die Lesezugriffe. Bei SSDs sind die Effekte der Defragmentierung weniger stark ausgeprägt.

► -C

zeigt einen Defragmentierungszähler für die betreffende(n) Datei(en) an, führt aber keine Änderungen durch. Wenn `e4defrag` mit dieser Option für das gesamte Dateisystem ausgeführt wird, liefert das Kommando nach geraumer Zeit eine Liste der Dateien, die am stärksten fragmentiert sind.

► -V

zeigt Informationen vor und nach der Defragmentierung jeder Datei an.

**echo** [optionen] zeichenkette

Das `bash`-Kommando `echo` gibt die angegebene Zeichenkette aus. Wenn die Zeichenkette Leer- oder Sonderzeichen enthält, muss sie in doppelte oder einfache Apostrophe eingeschlossen werden, je nachdem, ob Shell-Variablen ausgegeben werden sollen oder nicht.

► -e

interpretiert diverse Backslash-Zeichenkombinationen, z. B. `\a` als Signalton, `\n` als Zeilenende und `\t` als Tabulator (siehe auch `help echo`). Somit gibt `echo -e "\a"` einen Warnton aus.

► -n

wechselt beim Ende der Ausgabe nicht in eine neue Zeile. Die Ausgabe kann durch eine weitere `echo`-Anweisung fortgesetzt werden.

## Beispiel

Das folgende Kommando zeigt den aktuellen Inhalt der `$PATH`-Variablen an:

```
user$ echo "Die PATH-Variable enthält: $PATH"
```

**efibootmgr** [optionen]

In Linux-Systemen, die im EFI-Modus installiert und gestartet wurden, können über das Kommando `efibootmgr` diverse EFI-Einstellungen verändert werden. Diese Einstel-

lungen werden in einem nichtflüchtigen Speicher (NVRAM) des Mainboards gespeichert. `efibootmgr` dient primär dazu, neue EFI-Booteinträge einzurichten, vorhandene Einträge zu entfernen und die Standard-Bootreihenfolge festzulegen. Das Kommando setzt voraus, dass das Kernelmodul `efivars` geladen ist. Sollte das nicht der Fall sein, führen Sie `modprobe efivars` aus.

Beachten Sie, dass das `efivars`-Modul nur verwendet werden kann, wenn Linux im EFI-Modus gebootet wurde (nicht im BIOS-Modus!). Verwenden Sie für Reparaturarbeiten gegebenenfalls ein Linux-Live-System, das sich im EFI-Modus starten lässt.

► `-b n -B`

löscht den angegebenen EFI-Booteintrag. Die Nummern der EFI-Booteinträge ermitteln Sie, indem Sie `efibootmgr` ohne Parameter ausführen.

► `-c -l \EFI\distrib\grubdatei.efi -L name`

erzeugt einen neuen EFI-Booteintrag (`-c`) und gibt an, an welchem Ort sich die dazugehörige Bootloader-Datei befindet. Die Pfadangabe ist relativ zur EFI-Partition `/boot/efi`, und als Verzeichnistrenner muss `\` verwendet werden. `-L` gibt den Namen des EFI-Booteintrags an (üblicherweise den Distributionsnamen, standardmäßig `Linux`).

Der neue Eintrag wird automatisch zum Default-Eintrag. Optional können Sie mit `-p` die Nummer der EFI-Bootpartition (normalerweise 1) und mit `-d` den Device-Namen der ersten Festplatte/SSD angeben (normalerweise `/dev/sda`).

► `-n n`

bestimmt, welcher EFI-Booteintrag beim nächsten Neustart verwendet wird. Die Einstellung ist nicht bleibend, sondern gilt nur für den nächsten Rechnerstart.

► `-N`

löscht die mit `-n` festgelegte Einstellung für den nächsten Neustart.

► `-o n1,n2,n3,...`

legt die neue Reihenfolge der EFI-Booteinträge bleibend fest. Wenn Sie nur den Eintrag an der ersten Stelle festlegen möchten, reicht es aus, dessen Nummer anzugeben.

► `-p n`

legt fest, in welcher Partition sich die EFI-Bootpartition befindet (normalerweise in der ersten).

► `-q`

keine Ausgaben durchführen (*quiet*).

► -t

gibt an, wie lange EFI bei einem Neustart mit der Aktivierung des Default-Boot-eintrags warten soll (in Sekunden). Es hängt allerdings von der EFI-Implementierung ab, ob das EFI-Menü in dieser Wartezeit automatisch angezeigt wird. Auf einem meiner Testrechner wird das EFI-Menü generell nur angezeigt, wenn ich eine entsprechende Taste drücke (**F8**), Asus-Mainboard P8H67-M Evo).

► -T

löscht die mit -t eingestellte Wartezeiteinstellung, d. h., der Default-Booteintrag wird sofort aktiviert.

### Beispiel

Wenn das Kommando ohne weitere Optionen ausgeführt wird, listet es die EFI-Boot-einträge sowie einige weitere Parameter des EFI-Bootloaders auf:

```
root# efibootmgr
BootCurrent: 0000
Timeout: 1 seconds
BootOrder: 0000,0005,0003,0001,0002
Boot0000* ubuntu
Boot0001* Hard Drive
Boot0002* CD/DVD-Laufwerk
Boot0003* Windows Boot Manager
Boot0005* Fedora
```

Beim nächsten Neustart soll Fedora gestartet werden (abweichend von der gespeicherten Bootreihenfolge, in der Ubuntu an erster Stelle steht):

```
root# efibootmgr -n 5
```

Das nächste Kommando erzeugt einen neuen EFI-Booteintrag. Die Verdoppelung der \-Zeichen ist erforderlich, weil die Shell ein einfaches \-Zeichen zur Kennzeichnung von Sonderzeichen interpretiert.

```
root# efibootmgr -c -l \\EFI\\test\\abc.efi -L abc
```

### eject [name]

eject ohne Parameter wirft normalerweise die eingelegte CD/DVD aus. Falls die CD/DVD in das Dateisystem eingebunden war, führt eject vorher umount aus.

Bei Rechnern mit mehreren auswerfbaren Datenträgern (z. B. Magnetbändern) werden diese Möglichkeiten der Reihe nach getestet. Der erste gefundene Datenträger

wird ausgeworfen. Optional kann der Datenträger durch den Device-Namen oder das Mount-Verzeichnis angegeben werden.

```
enscript [optionen] quelldatei -p zieldatei
```

Das Kommando `enscript` aus dem gleichnamigen Paket wandelt eine Textdatei wahlweise in die Formate PostScript, HTML oder RTF um.

- ▶ `--color`  
Syntaxhervorhebung mit Farben durchführen (muss mit `-E` kombiniert werden).
- ▶ `-E`  
führt bei Programmcode eine Syntaxhervorhebung durch (fett, kursiv).
- ▶ `-f font`  
verwendet die angegebene Schriftart (standardmäßig Courier in einer Größe von 10 Punkt).
- ▶ `-M papierformat`  
verwendet das angegebene Papierformat (z. B. A3, A4 oder Letter).
- ▶ `--n`  
formatiert den Text mit *n* Spalten. `-2` bewirkt somit eine zweispaltige Formatierung.
- ▶ `-r` bzw. `--landscape`  
füllt das Blatt im Querformat.
- ▶ `-W format` bzw. `--language=format`  
gibt das gewünschte Format an. Zur Auswahl stehen unter anderem PostScript (gilt standardmäßig), `html` und `rtf`.
- ▶ `-X charset`  
gibt den Zeichensatz des Texts an (z. B. `ascii`, `latin1`). Unicode wird leider nicht unterstützt. Um Unicode-Texte in das PostScript-Format zu konvertieren, verwenden Sie das Kommando `paps`.

```
epstopdf [optionen] datei.eps
```

epstopdf wandelt die EPS-Datei in eine PDF-Datei um und speichert das Ergebnis unter dem Namen datei.pdf.

- ▶ `--exact`  
wertet die ExactBoundingBox aus (anstatt der normalen BoundingBox).
- ▶ `--hires`  
wertet die HiresBoundingBox aus (anstatt der normalen BoundingBox).
- ▶ `--nocompress`  
verzichtet auf eine Komprimierung der PDF-Daten.
- ▶ `--outfile=name`  
speichert das Ergebnis unter dem angegebenen Dateinamen.

```
ethtool [optionen] device [parameter]
```

ethtool ermittelt bzw. verändert (Hardware-)Parameter von Ethernet-Adaptern. Wenn Sie an das Kommando einfach einen Schnittstellennamen ohne weitere Optionen übergeben, erhalten Sie eine Auflistung der wichtigsten Eckdaten. Dazu zählen die unterstützten Ports, die Link-Modi, die Geschwindigkeit, der Duplex-Modus etc.

- ▶ `-p device n`  
schaltet für *n* Sekunden die LED der Schnittstelle ein. Bei Servern mit mehreren Adaptern vereinfacht das die Identifizierung der Steckerbuchse. Bei manchen Adaptern blinkt die LED dann für die angegebene Zeit. Andere Adapter unterstützen gar keine LED-Steuerung – dann kommt es zur Fehlermeldung *operation not supported*.
- ▶ `-s device para1 value1 para2 value2 ...`  
verändert mehrere Parameter der Schnittstelle. Zulässige Parameternamen sind unter anderem speed, duplex, port, mdix, autoneg und advertise.

## Beispiel

Das folgende Kommando stellt die Geschwindigkeit des Netzwerkadapters explizit mit 100 MBit/Sekunde ein, aktiviert den Duplex-Modus und deaktiviert die Auto-Negotiate-Funktion:

```
root# ethtool -s eth0 speed 100 duplex full autoneg off
```

```
eval $var
```

Das bash-Kommando `eval` interpretiert den Inhalt der Variablen als Kommandozeile, wertet diese Zeile mit allen bekannten Substitutionsmechanismen aus und führt das Kommando schließlich aus. `eval` ist immer dann erforderlich, wenn ein in einer Variablen gespeichertes Kommando ausgeführt werden soll und dieses Kommando diverse Sonderzeichen der Shell enthält.

### Beispiel

Erst mit der Verwendung von `eval` kann das in der Variablen `kom` gespeicherte Kommando ausgeführt werden. Der erste Versuch, das Kommando auszuführen, scheitert, weil die bash das Pipe-Zeichen `|` nicht mehr auswertet, nachdem sie `$kom` durch seinen Inhalt ersetzt hat.

```
user$ kom="ls | more"
user$ $kom
ls: |: No such file or directory
ls: more: No such file or directory
user$ eval $kom
```

```
exec kommando
```

Das bash-Kommando `exec` startet das im Parameter angegebene Kommando als Ersatz zur laufenden Shell. Das Kommando kann beispielsweise dazu verwendet werden, eine andere Shell zu starten. Die laufende Shell wird dadurch beendet. Bei einem normalen Kommandostart ohne `exec` läuft die Shell hingegen weiter und nimmt nach dem Ende des Kommandos weitere Eingaben entgegen bzw. setzt das laufende Script fort.

```
exfatlabel device [label]
```

`exfatlabel` liest den Namen eines exFAT-Dateisystems aus bzw. verändert diesen. Das Kommando befindet sich bei vielen Distributionen im Paket `exfat-utils`. Zum Einrichten eines exFAT-Dateisystems verwenden Sie [`mkfs.exfat`](#).

```
exit [rückgabewert]
```

Das bash-Kommando `exit` beendet ein Shell-Skript bzw. die laufende Shell. Wenn kein Rückgabewert angegeben wird, gibt die Shell 0 zurück, also »OK«.



**expand** [optionen] datei

expand ersetzt in der angegebenen Textdatei alle Tabulator-Zeichen durch eine entsprechende Zahl von Leerzeichen. Wenn keine Optionen angegeben werden, nimmt expand einen Tabulator-Abstand von acht Zeichen an. Die Ergebnis wird zur Standardausgabe geleitet.

► -n

verändert den Tabulator-Abstand auf *n* Zeichen.

### Beispiel

Das folgende Kommando entfernt die Tabulatorzeichen in `test.java` und speichert das Ergebnis in `test-no-tabs.java`. Die ursprüngliche Datei wurde mit einem Editor verfasst, der Tabulatorzeichen mit vier Leerzeichen gleichsetzte.

```
user$ expand -n 4 test.java > test-no-tabs.java
```

**export** [optionen] variable [=wert]

Das bash-Kommando `export` deklariert die angegebene Shell-Variable als Umgebungsvariable. Damit ist die Variable auch in allen aufgerufenen Kommandos und Subshells verfügbar. Optional kann dabei auch eine Variablenzuweisung erfolgen. Wenn das Kommando ohne Parameter aufgerufen wird, werden alle Umgebungsvariablen angezeigt.

► -n

macht eine Umgebungsvariable wieder zu einer normalen Shell-Variablen. Das Kommando hat damit genau die umgekehrte Wirkung wie bei der Verwendung ohne Optionen.

**exportfs** [optionen]

Das Kommando `exportfs -a` meldet Änderungen in der NFS-Konfigurationsdatei `/etc/exports` an den laufenden NFS-Server. Das Kommando muss also ausgeführt werden, damit in `/etc/exportfs` durchgeführte Änderungen wirksam werden.

```
expr ausdruck
expr zeichenkette : muster
```

`expr` wertet den angegebenen Ausdruck arithmetisch aus oder führt einen Mustervergleich für Zeichenketten aus. Zwischen den angegebenen Variablen, Zahlen und Operatoren müssen jeweils Leerzeichen stehen! Eine kurze Beschreibung aller zulässigen Operatoren finden Sie in den `man`-Seiten zu `expr`. Beachten Sie, dass viele Operatorzeichen in der Shell durch das Zeichen `\` vor der unmittelbaren Auswertung geschützt werden müssen.

Wenn Sie die `bash` als Shell verwenden, können Sie darin auch ohne `expr` rechnen: Arithmetische Ausdrücke können dort in der Form `$(ausdruck)` angegeben werden. Im Muster können die bei [grep](#) beschriebenen Jokerzeichen verwendet werden. Das Muster muss normalerweise in Apostrophe gestellt werden, um eine Auswertung der Sonderzeichen durch die Shell zu vermeiden.

### Beispiele

Das erste `expr`-Kommando führt eine einfache Berechnung durch. Das zweite Kommando liefert die Anzahl der Zeichen, die dem Musterausdruck maximal entsprechen. Das dritte Kommando extrahiert den geklammerten Teil des Musters aus einer Zeichenkette.

```
user$ expr 3 + 7 \* 19
136
user$ expr abcdefghi : 'a.*g'
7
user$ expr abc_efg_hij : '.*_\(.*\)_.*'
efg
```

```
fc-list [muster]
```

`fc-list` listet unter `X` alle skalierbaren Schriftarten auf, die dem (optionalen) Suchmuster entsprechen. Das Ergebnis kann mit `| sort` sortiert werden.

```
fdisk [optionen] [device]
```

`fdisk` ist ein interaktives Programm zur Partitionierung von Festplatten. Wenn beim Start keine Device-Datei angegeben wird, bezieht sich `fdisk` automatisch auf die erste Festplatte. `fdisk` kann nur Datenträger mit MBR-Partitionstabellen bearbeiten. Wenn die Festplatte oder SSD eine GUID-Partitionstabelle (GPT) enthält, müssen Sie anstatt von `fdisk` die Kommandos [gdisk](#) oder [parted](#) zur Partitionierung verwenden.

## ► -c

deaktiviert den DOS-Kompatibilitätsmodus. Bei aktuellen `fdisk`-Optionen gilt diese Option standardmäßig. Die Option ist unbedingt erforderlich, wenn Sie eine neue Festplatte mit 4-kByte-Sektoren partitionieren!

## ► -l

listet alle Partitionen aller Festplatten auf und beendet `fdisk` anschließend.

► -s *device*

liefert die Größe der angegebenen Partitionen in Blöcken und beendet `fdisk` anschließend.

## ► -u

verwendet für Größenangaben Sektoren (in der Regel zu 512 Byte) anstatt von Zylindern. Bei aktuellen `fdisk`-Versionen gilt diese Option standardmäßig. Das ist insofern zweckmäßig, als die von Festplatten gelieferten Zylinderangaben schon lange nicht mehr dem realen Aufbau von Festplatten entsprechen.

## Tastaturkommandos

`fdisk` wird interaktiv durch Tastaturkommandos gesteuert (siehe [Tabelle 11](#) im Abschnitt *Tastenkürzel*). Sämtliche Änderungen werden erst mit dem Kommando `[W]` (*write*) ausgeführt. Vorher können Sie mit `[V]` (*verify*) überprüfen, ob alle internen Informationen mit der Platte übereinstimmen. Das ist eine zusätzliche Sicherheitskontrolle. Normalerweise besteht die Reaktion auf `[V]` nur darin, dass die Anzahl der 512-Byte-Sektoren angezeigt wird, die von keiner primären oder logischen Partition erfasst und somit noch ungenutzt sind. Wenn Sie sich unsicher sind, können Sie `fdisk` jederzeit mit `[Q]` (*quit*) oder auch mit `[Strg]+[C]` verlassen – Ihr Datenträger bleibt dann so, wie er momentan ist.

## Beispiel

Die folgenden Zeilen zeigen, wie auf einer 1,5 TByte großen Festplatte mit zwei Partitionen eine dritte, 30 GByte große Partition eingerichtet wird. `[P]` listet die Partitionen auf, `[N]` erzeugt eine neue Partition, `[W]` speichert die Änderungen.

```
root# fdisk -c -u /dev/sda
Befehl (m für Hilfe): p
Platte /dev/sda: 1500.3 GByte, 1500301910016 Byte
255 Köpfe, 63 Sektoren/Spur, 182401 Zylinder, zusammen 2930277168 Sektoren
Einheiten = Sektoren von 1 × 512 = 512 Bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

Disk identifier: 0x0004b057

Gerät	boot.	Anfang	Ende	Blöcke	Id	System
/dev/sda1	*	2048	97656831	48827392	83	Linux
/dev/sda2		97656832	101562367	1952768	82	Linux Swap / Solaris

Befehl (m für Hilfe): **n**

Befehl Aktion

e Erweiterte

p Primäre Partition (1-4)

**p**

Partitionsnummer (1-4): **3**

Erster Sektor (101562368-2930277167, Vorgabe: 101562368): **<return>**

Benutze den Standardwert 101562368

Last Sektor, +Sektoren or +sizeK,M,G (101562368-2930277167,  
Vorgabe: 2930277167): **+30G**

Befehl (m für Hilfe): **p**

Gerät	boot.	Anfang	Ende	Blöcke	Id	System
/dev/sda1	*	2048	97656831	48827392	83	Linux
/dev/sda2		97656832	101562367	1952768	82	Linux Swap / Solaris
/dev/sda3		101562368	164476927	31457280	83	Linux

Command (m for help): **w**

Die Partitionstabelle wurde verändert!

**ffmpeg** [inopts] [-i infile] [outopts] outfile

Das Kommando **ffmpeg** aus dem gleichnamigen Paket konvertiert Video-Dateien von einem Format in ein anderes. Bei der Angabe von Optionen ist zu beachten, dass diese nur für die als Nächstes angegebene Video-Datei gelten. Die Reihenfolge der Optionen ist daher entscheidend für die korrekte Funktion des Kommandos. Sofern Sie keine abweichenden Einstellungen vornehmen, verwendet **ffmpeg** für die Ergebnisdatte dieselben Codecs und Einstellungen wie in der Quelldatei.

Neben den hier aufgezählten Optionen gibt es unzählige weitere. Die optimale Einstellung der Optionen ist eine Wissenschaft für sich. Wenn Sie sich damit nicht auseinandersetzen möchten, ist es empfehlenswert, zur Video-Recodierung eine grafische Benutzeroberfläche einzusetzen, z. B. HandBrake oder traGtor.

► **-acodec/-vcodec type**

legt den gewünschten Audio- bzw. Video-Codec fest. Eine Liste der verfügbaren Codecs liefert **ffmpeg -codecs**.

► **-b n**

gibt die gewünschte Bitrate in Bits pro Sekunde an (standardmäßig 200 kBit/s).

- ▶ `-f format`  
verwendet das angegebene Format.
- ▶ `-formats`  
listet die unterstützten Formate, Codecs, Filter etc. auf.
- ▶ `-r n`  
gibt die gewünschte Frame-Rate an (standardmäßig 25 Frames pro Sekunde).
- ▶ `-s size`  
gibt die gewünschte Frame-Größe in der Form `wxh` an (z. B. 800x600). Alternativ können Sie eine vordefinierte Zeichenkette angeben (z. B. `vga` oder `hd720`, siehe man `ffmpeg`).
- ▶ `-ss position`  
startet die Konvertierung zum angegebenen Zeitpunkt. Die Zeitangabe erfolgt wahlweise in Sekunden oder in der Form `hh:mm:ss[.xxx]`.
- ▶ `-t duration`  
konvertiert nur die angegebene Zeitspanne (und nicht die gesamte Datei). Die Angabe der Zeitspanne erfolgt wie bei der Option `-ss`.
- ▶ `-target type`  
gibt das gewünschte Zielformat an (z. B. `vcd`, `dvd` etc.).
- ▶ `-vpre quality`  
wählt eine Datei mit vordefinierten Optionen (Presets) für den angegebenen Codec aus (z. B. `default`, `medium`, `max` oder `fast`). Die zur Auswahl stehenden Preset-Dateien befinden sich üblicherweise im Verzeichnis `/usr/share/ffmpeg`.
- ▶ `-y`  
überschreibt bereits vorhandene Dateien.

### Beispiel

Das folgende Kommando erstellt eine Filmdatei in DVD-Auflösung:

```
user$ ffmpeg -i in.avi -target pal-dvd -y out.avi
```

Das zweite Beispiel wandelt eine OGV-Filmdatei in eine YouTube-taugliche MPEG-Datei um:

```
user$ ffmpeg -i in.ogv -vcodec libx264 -vpre medium -acodec copy -y out.mpg
```

### **fg** [prozess]

Das bash-Kommando `fg` setzt einen mit `[Strg]+[Z]` unterbrochenen Prozess im Vordergrund fort. Wenn kein Prozess angegeben wird, gilt `fg` für den zuletzt unterbrochenen bzw. für den zuletzt im Hintergrund gestarteten Prozess. Andernfalls muss der Prozess durch seinen Namen oder durch die bash-interne Jobnummer (nicht PID) angegeben werden. Das Kommando kann auch in der Kurzschreibweise `%prozess` verwendet werden.

### **fgconsole** [--next-available]

`fgconsole` gibt die Nummer der gerade aktiven Konsole aus. Mit der Option `--next-available` ermittelt das Kommando die Nummer der ersten freien (ungenutzten) Konsole. Zum Wechsel der aktiven Konsole verwenden Sie das Kommando `chvt`.

### **file** [optionen] datei

`file` versucht festzustellen, welchen Dateityp die angegebene Datei hat. Als Ergebnis liefert `file` eine Zeichenkette mit dem Dateinamen und dem Typ der Datei. `file` wertet nicht den Dateinamen bzw. dessen Kennung aus, sondern den Inhalt der Datei!

#### ► -Z

versucht, den Datentyp einer komprimierten Datei zu erkennen.

### Beispiel

`name.jpg` enthält, wie der Dateiname erwarten lässt, eine JPEG-Bitmap-Datei:

```
user$ file name.jpg
name.jpg: JPEG image data, JFIF standard 1.02
```

### **find** [pfadangabe] [suchoptionen]

`find` hilft bei der Suche nach Dateien. Dabei können verschiedene Suchkriterien (Muster für den Dateinamen, Dateigröße, Datum der Erstellung oder des letzten Zugriffs etc.) bei der Suche berücksichtigt werden. Es besteht sogar die Möglichkeit, auf alle Dateien, die diese Bedingungen erfüllen, mit einem anderen Programm (etwa `grep`) weitere Selektionskriterien anzuwenden. Auf diese Weise könnten beispielsweise alle `*.tex`-Dateien gefunden werden, die in den letzten drei Tagen bearbeitet wurden und die den Text »Grafikprogrammierung« enthalten. Aufgrund der großen Anzahl

möglicher Suchkriterien liefert man `find` eine viele Seiten lange Beschreibung des Kommandos.

Die folgende Syntaxbeschreibung bezieht sich explizit auf die unter Linux gebräuchliche GNU-`find`-Implementierung. Sie zeichnet sich durch einige kleinere Syntaxunterschiede gegenüber den `find`-Varianten anderer Unix-Systeme aus: Dort *muss* eine Pfadangabe erfolgen, während GNU-`find` das aktuelle Verzeichnis als Startverzeichnis verwendet. Außerdem verlangen die meisten `find`-Implementierungen den Einsatz der Option `-print` zur Anzeige der Suchergebnisse, was bei GNU-`find` nicht erforderlich ist.

### Allgemeine Optionen

Im Gegensatz zu den meisten anderen Kommandos durchsucht `find` automatisch alle Unterverzeichnisse. Wenn das nicht erwünscht ist, muss die Anzahl der Unterverzeichnisse durch `-maxdepth` eingeschränkt werden.

► `-depth`

bearbeitet zuerst das aktuelle Verzeichnis und erst dann die Unterverzeichnisse. Je nachdem, wo Sie die gesuchte Datei vermuten, führt diese Vorgehensweise erheblich schneller zu ersten Ergebnissen.

► `-follow`

bearbeitet auch Verzeichnisse, die durch symbolische Links erfasst werden.

► `-maxdepth n`

schränkt die Suche auf *n* Verzeichnisebenen ein. Mit `-maxdepth 1` werden überhaupt keine Unterverzeichnisse berücksichtigt.

### Suchkriterien

Es können mehrere Suchkriterien gleichzeitig genannt werden. Diese Kriterien werden logisch mit UND verknüpft. Die Suche wird abgebrochen, sobald das erste Kriterium nicht erfüllt ist – die Reihenfolge der Kriterien kann also Einfluss auf die Geschwindigkeit des Kommandos haben. Kriterien können mit `\(` und `\)` gruppiert, mit `!` negiert und mit `-o` logisch ODER-verknüpft werden.

► `-executable`

findet Dateien, die ausführbar sind, bei denen also das Execute-Bit gesetzt ist.

► `-group gruppenname` oder `-nogroup gruppenname`

findet Dateien, die der angegebenen Gruppe angehören (bzw. die ihr nicht angehören).

- ▶ `-mmin n`  
funktioniert wie `mtime`, allerdings erfolgt die Zeitangabe in Minuten.
- ▶ `-mtime n`  
findet Dateien, deren Inhalt zuletzt vor genau *n* Tagen verändert wurde. Wenn vor der Zahl ein `+` angegeben wird, dann werden alle Dateien erfasst, die älter als *n* Tage sind. Ein vorangestelltes `-` liefert Dateien, die jünger als *n* Tage sind. `-mtime 0` liefert Dateien, die in den letzten 24 Stunden verändert wurden.
- ▶ `-name suchmuster`  
findet Dateien, die dem angegebenen Suchmuster entsprechen. Wenn das Suchmuster Jokerzeichen enthält, muss es in Apostrophe gestellt werden. Wenn `find` nicht zwischen Groß- und Kleinschreibung unterscheiden soll, verwenden Sie statt `-name` die Option `-iname`.
- ▶ `-newer referenzdatei`  
findet Dateien, die neuer sind als die Referenzdatei, die also nach der Referenzdatei erzeugt oder verändert wurden. Sie können beispielsweise beim Beginn eines Backups mit `touch` eine `timestamp`-Datei erzeugen. Bei einem inkrementellen weiteren Backup zu einem späteren Zeitpunkt berücksichtigen Sie dann nur die Dateien, für die `-newer timestamp` gilt.
- ▶ `-path suchmuster`  
findet Dateien, die dem angegebenen Suchmuster entsprechen. Die Option geht über `-name` hinaus, weil das Suchmuster jetzt nicht nur den Dateinamen, sondern auch den Pfad dorthin betrifft. Diese Option ist flexibler als die direkte Pfadangabe im ersten Argument von `find`, weil hier die Jokerzeichen auch das Zeichen `/` erfassen.
- ▶ `-perm zugriffsbits`  
findet Dateien, deren Zugriffsbits genau dem angegebenen Oktalwert (siehe `chmod`) entsprechen. Wenn dem Oktalwert ein `-` vorangestellt wird, dann reicht es, wenn die Datei mindestens eines dieser Zugriffsbits enthält. Wenn ein `+` vorangestellt wird, darf die Datei auch darüber hinausgehende Zugriffsrechte haben.
- ▶ `-size dateigröße`  
gibt die gewünschte Dateigröße vor. Die Angabe erfolgt standardmäßig in Vielfachen von 512. `3` bezeichnet also Dateien zwischen 1024 und 1536 Bytes. Durch die zusätzlichen Zeichen `c` oder `k` kann die Größe in Byte oder kByte angegeben



werden. Ein vorangestelltes + erfasst alle größeren Dateien, ein vorangestelltes - alle kleineren Dateien. `-size +10k` liefert daher alle Dateien, die größer als 10 kByte sind.

► `-type zeichen`

schränkt die Suche auf bestimmte Dateitypen ein. Die wichtigsten Zeichen sind `f` für reguläre Dateien, `d` für Verzeichnisse (*directories*) und `l` für symbolische Links.

► `-user username` oder `-nouser username`

findet Dateien, die dem angegebenen Benutzer gehören (bzw. die ihm nicht gehören).

### Aktionen beim Finden einer Datei

► `-exec kommando [optionen] {} \;`

ruft das angegebene Kommando auf und übergibt den Dateinamen der gefundenen Datei, die alle bisher verarbeiteten Kriterien erfüllt hat. Das Kommando kann nun einen Test durchführen, ob die Datei weiteren Kriterien entspricht. Ein typisches Programm, das durch `-exec` aufgerufen wird, ist `grep`. `{}` steht als Platzhalter für den Dateinamen. `;` schließt den Kommandoaufruf ab, d. h., dahinter können weitere `find`-Optionen angegeben werden. `\` ist innerhalb der Shell erforderlich, um die Interpretation von `;` als Sonderzeichen zu verhindern.

► `-ls`

liefert detaillierte Angaben zu jeder gefundenen Datei (so wie `ls -l`).

► `-print`

gibt die gefundenen Dateinamen auf dem Bildschirm aus. Diese Option ist die Standardeinstellung, sofern nicht `-exec` verwendet wird.

► `-print0`

trennt die Dateinamen der Ergebnisliste durch 0-Bytes. Das ermöglicht eine Weiterverarbeitung von Dateinamen mit Leerzeichen durch `xargs`.

► `-printf format`

gibt die gefundenen Dateinamen und andere Informationen auf dem Bildschirm aus. In der Formatzeichenkette kann angegeben werden, in welcher Form die Ausgabe erfolgt und welche Zusatzinformationen mit ausgegeben werden (etwa die Dateigröße, das Datum der letzten Änderung etc.). Die Syntax für die Formatzeichenkette ist in den `man`-Seiten beschrieben.

## Beispiele

Das erste `find`-Kommando liefert alle Dateien im aktuellen Verzeichnis (auch versteckte Dateien, aber keine Dateien aus Unterverzeichnissen); das zweite Kommando liefert nur gewöhnliche Dateien (aber keine unsichtbaren Dateien):

```
user$ find -maxdepth 1 -type f -name '*'
user$ find -maxdepth 1 -type f -name '[!.]*'
```

Das folgende Kommando durchsucht das `/etc`-Verzeichnis nach Dateien, die in den letzten zwei Wochen verändert wurden. Das Kommando muss mit `root`-Rechten ausgeführt werden, weil einige Dateien in `/etc` nicht von gewöhnlichen Benutzern gelesen werden dürfen:

```
root# sudo find /etc -mtime -14
```

Das folgende Kommando löscht alle Backup-Dateien, die durch das Zeichen `~` am Ende des Dateinamens gekennzeichnet sind. `find` berücksichtigt dabei auch alle Unterverzeichnisse. Die Liste der zu löschenden Dateien wird durch Kommandosubstitution `$(kommando)` an `rm` weitergeleitet.

```
root# rm $(find . -name '*~')
```

Falls es sich um sehr viele Dateien handelt, tritt bei der Ausführung des obigen Kommandos ein Fehler auf: Die Kommandozeile mit allen `*~`-Dateien wird so lang, dass sie die maximale Kommandozeilenlänge überschreitet. In solchen Fällen müssen Sie entweder die `-exec`-Option des `find`-Kommandos oder das Kommando `xargs` zu Hilfe nehmen. Die `xargs`-Variante hat den Vorteil, dass sie auch mit Dateinamen zurechtkommt, die Leerzeichen enthalten.

```
user$ find -name '*~.jpg' -exec rm {} \;
```

Weitere `find`-Beispiele finden Sie bei der Beschreibung der Kommandos `grep` und `xargs`.

```
findmnt [optionen] [device/mountpoint]
```

`findmnt` ohne Parameter liefert eine baumförmige Liste aller im Verzeichnisbaum integrierten Dateisysteme. Das Ergebnis enthält mehr Einträge als bei `mount` oder `df`, weil `findmnt` auch diverse `/dev`-, `proc`-, `sys`- und `/run`-Dateisysteme zum internen Informationsaustausch berücksichtigt, also sogenannte Pseudo-Dateisysteme.

Durch die Angabe eines Devices oder `mount`-Punktes reduziert sich die Liste auf alle so ausgewählten Dateisysteme. Weitere zulässige Angaben sind die UUID oder der Name (Label) eines Dateisystems bzw. die Major- und Minor-Nummer des Devices in der Schreibweise `major:minor`.

- ▶ `-A`  
zeigt alle Dateisysteme an.
- ▶ `-l`  
zeigt das Ergebnis als Tabelle an (nicht als Baum).
- ▶ `--mtab` bzw. `--fstab`  
berücksichtigt nur die Einträge von `/etc/mtab` bzw. `/etc/fstab`. Standardmäßig berücksichtigt findmnt die Mount-Tabelle des Kernels in der Pseudo-Datei `/proc/self/mountinfo`.
- ▶ `-o spalten` bzw. `--output spalten`  
gibt nur die in `spalten` angegebenen Informationen spaltenweise aus. Zulässige Spaltencodes sind unter anderem:  
  
SOURCE: Device  
TARGET: mount-Pfad  
FSTYPE: Dateisystemtyp  
OPTIONS: alle mount-Optionen  
SIZE: Größe des Dateisystems  
  
Weitere Codes liefert findmnt `--help`. Die Spaltencodes werden durch Kommas getrennt, beispielsweise `-o SOURCE,TARGET,FSTYPE`. Leerzeichen sind nicht zulässig!
- ▶ `-t typliste` bzw. `--types typliste`  
liefert nur Dateisysteme des angegebenen Typs. Mit `-t ext2,ext3,ext4` werden somit nur ext-Dateisysteme in den Versionen 2, 3 und 4 berücksichtigt.

## Beispiel

findmnt zeigt alle aktiven Dateisysteme des Testrechners:

```
root#
TARGET                                SOURCE      FSTYPE      OPTIONS
/                                     /dev/mapper/fedora-root
                                     ext4        rw,relatime,...
/proc                                 proc        proc        rw,nosuid,...
/proc/sys/fs/binfmt_misc             systemd-1   autofs      rw,relatime,...
/sys                                 sysfs       sysfs       rw,nosuid,...
/sys/kernel/security                 securityfs  securityfs  rw,nosuid,...
/sys/fs/selinux                      selinuxfs   selinuxfs   rw,relatime
...
```

### firewall-cmd optionen

firewall-cmd liest bzw. verändert die Firewall-Konfiguration unter Fedora sowie unter CentOS und RHEL ab Version 7. firewall-cmd kommuniziert mit dem im Hintergrund laufenden Programm firewalld. Dieses Programm ordnet jeder Netzwerkschnittstelle eine Zone zu. Eine »Zone« im Sinne von firewalld ist eine Sammlung von Regeln für einen bestimmten Anwendungszweck. Die folgende Aufzählung beschreibt die wichtigsten Zonen:

- ▶ `block`  
blockiert jeden Netzwerkverkehr; der Absender erhält eine ICMP-Fehlermeldung.
- ▶ `drop`  
blockiert jeden Netzwerkverkehr; der Absender wird nicht informiert.
- ▶ `external`  
blockiert die meisten Ports und aktiviert Masquerading (IPv4). Bei einem Router ist diese Zone für die Schnittstelle vorgesehen, die die Verbindung zum Internet herstellt.
- ▶ `FedoraWorkstation` und `FedoraServer`  
kommen standardmäßig für Netzwerkschnittstellen von Fedora-Installationen zum Einsatz. `FedoraServer` blockiert alle Dienste außer SSH und DHCP-Client. `FedoraWorkstation` ist deutlich liberaler und akzeptiert auch Samba-Client-Verbindungen sowie den Verkehr über alle Ports zwischen 1025 und 65535.
- ▶ `home` und `internal`  
blockiert die meisten Ports, akzeptiert aber Samba (nur als Client), CUPS und Zeroconf/Avahi/mdns. Beide Zonen sind für Rechner in einem als einigermaßen sicher geltenden lokalen Netzwerk gedacht. Wenn Sie diese Zone nutzen und selbst Windows-Netzwerkverzeichnisse freigeben möchten, müssen Sie außerdem den Dienst SAMBA freischalten.
- ▶ `public`  
ähnlich wie `home`, blockiert aber auch CUPS und Samba-Client-Funktionen. Die Zone ist für die Internet-Nutzung in unsicheren Netzwerken gedacht, z.B. in einem öffentlichen WLAN.
- ▶ `trusted`  
erlaubt jeden Netzwerkverkehr. Diese Zone ist für gut gesicherte lokale Netzwerke gedacht, aber nicht für WLAN-Verbindungen.

firewall-cmd führt die Änderungen normalerweise nur dynamisch durch. Wenn Änderungen dauerhaft gelten sollen, müssen Sie die Option `--permanent` angeben. Die Regeln werden dann im Verzeichnis `/etc/firewalld` gespeichert.

## Optionen

- ▶ `--add-interface=name` bzw. `--remove-interface=name`  
ordnet die angegebene Schnittstelle der Default-Zone oder der durch die Option `--zone` angegebenen Zone zu bzw. löst diese Zuordnung wieder. Bevor eine Schnittstelle einer neuen Zone zugeordnet werden kann, muss die bisherige Zuordnung explizit entfernt werden!
- ▶ `--add-prot=name` bzw. `--remove-port=name`  
erlaubt die Nutzung eines TCP-Ports in der mit `--zone` angegebenen Zone bzw. blockiert den Port.
- ▶ `--add-service=name` bzw. `--remove-service=name`  
erlaubt die Nutzung eines Netzwerkdiensts (z. B. https oder ssh) in der mit `--zone` angegebenen Zone bzw. blockiert den Dienst.
- ▶ `--get-active-zones`  
listet alle Zonen auf, die momentan in Verwendung sind, und gibt an, welche Netzwerkschnittstellen den Zonen zugeordnet sind.
- ▶ `--get-default-zone`  
ermittelt die Default-Zone.
- ▶ `--get-zones`  
listet alle Zonen auf, die firewall-cmd kennt.
- ▶ `--get-zone-of-interface=name`  
ermittelt die Zone, die der angegebenen Netzwerkschnittstelle zugeordnet ist.
- ▶ `--list-all-zones`  
liefert die Eckdaten aller definierten Firewall-Zonen und gibt an, welche Netzwerkschnittstellen den Zonen zugeordnet sind.
- ▶ `--permanent`  
speichert Änderungen an der Firewall-Konfiguration bleibend. Die Änderungen werden aber nicht direkt angewendet. Wenn Sie eine Änderung sofort *und* bleibend durchführen möchten, müssen Sie das betreffende Kommando *zweimal* ausführen: einmal mit `--permanent` und einmal ohne diese Option.

► `--set-default-zone=name`

definiert die Default-Zone. Diese gilt für alle Netzwerkschnittstellen, denen nicht explizit eine andere Zone zugewiesen wird.

► `--state`

ermittelt, ob `firewalld` läuft oder nicht.

► `--zone=name`

gibt an, welche Zone verändert, bearbeitet oder gelesen werden soll.

### Beispiele

Die folgenden Kommandos ermitteln zuerst, welcher Zone die Netzwerkschnittstelle `enp0s3` zugeordnet ist, und ordnet die Schnittstelle dann der Zone `trusted` zu. Damit können sämtliche Dienste über diese Schnittstelle genutzt werden. Die Änderungen sind nicht permanent, gelten also nur bis zum nächsten Neustart des Rechners bzw. des Firewall-Dämons.

```
root# firewall-cmd --get-zone-of-interface=enp0s3
public
root# firewall-cmd --zone=public --remove-interface=enp0s3
success
root# firewall-cmd --zone=trusted --add-interface=enp0s3
success
```

Beim zweiten Beispiel dient das erste Kommando dazu, herauszufinden, welcher Zone die Netzwerkschnittstelle `enp0s5` zugeordnet ist. Über diese Schnittstelle soll der Rechner einen Webserver nach außen anbieten. Aber anstatt die Schnittstelle generell der Zone `trusted` zuzuordnen und so auch alle anderen erdenklichen Dienste freizuschalten, werden jetzt die Regeln für die Zone geändert (hier `FedoraWorkstation`), und das dauerhaft: Diese Zone erlaubt jetzt die Protokolle HTTP und HTTPS. Das `reload`-Kommando aktiviert die Änderungen an der Zone.

```
root# firewall-cmd --get-zone-of-interface=enp0s5    (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=http
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=https
root# firewall-cmd --reload
```

**fold** [optionen] datei

`fold` umbricht Textzeilen bei einer Länge von 80 Zeichen und zeigt das Ergebnis auf dem Bildschirm an.

► `-s` bzw. `--spaces`

versucht, den Umbruch an der Stelle eines Leerzeichens (also zwischen zwei Wörtern) durchzuführen. Die Zeilen werden damit kürzer oder gleich  $n$  Zeichen.

► `-w n` bzw. `--width n`

stellt eine maximale Textbreite von  $n$  Zeichen ein. Ohne diese Option gelten standardmäßig 80 Zeichen.

```
for var [in liste;] do
    kommandos
done
```

`for` bildet Schleifen in `bash`-Scripts. In die angegebene Variable werden der Reihe nach alle Listenelemente eingesetzt. Die Liste kann auch mit Jokerzeichen für Dateinamen oder mit `{...}`-Elementen zum Zusammensetzen von Dateinamen gebildet werden. Wenn auf die Angabe der Liste verzichtet wird, durchläuft die Variable alle Parameter, die der Shell-Datei übergebenen wurden (also in `$*`).

### Beispiel

`for` kann nicht nur in Shell-Scripts verwendet werden, sondern auch im Terminal, um einige Kommandos auf eine Reihe von Dateien anzuwenden. Das folgende Kommando gibt die Dateinamen aller `*.jpg`-Dateien aus. Sie können nun `echo` durch ein anderes Kommando ersetzen, das die `*.jpg`-Dateien druckt, skaliert etc.

```
user$ for i in *.jpg; do echo $i; done
```

Ein weiteres `for`-Beispiel finden Sie bei der Beschreibung des `cp`-Kommandos.

```
free [optionen]
```

`free` zeigt an, wie der verfügbare Speicherplatz (RAM und Swap-Speicher) genutzt ist. Als Einheit werden normalerweise `kByte` verwendet. Mit der Option `-m` rechnet `free` in `MByte`.

### Beispiel

Im folgenden Beispiel ist scheinbar beinahe der gesamte Speicher (1 GByte) genutzt. Tatsächlich werden aber rund 330 MByte vom Betriebssystem zur Zwischenspeicherung von Dateien verwendet, sodass insgesamt noch ca. 420 MByte für Anwendungs-

programme bereitstehen (zweite Zeile). Der Swap-Speicherplatz wird bis auf wenige MByte nicht genutzt.

```
user  free
      total      used      free      shared  buffers  cached
Mem:   1024008    933712    90296         0     45744    284836
-/+ buffers/cache:    603132    420876
Swap:   407544     20268    387276
```

### **fsck** [optionen] device

**fsck** überprüft die Konsistenz des Dateisystems und führt gegebenenfalls Reparaturen durch. Es darf nur von **root** ausgeführt werden. Je nach Typ des Dateisystems ruft **fsck** die Programme **fsck.ext3**, **fsck.ext4** etc. auf. Lesen Sie vor der Ausführung von **fsck** unbedingt die **man**-Seiten dieses und des dateisystemspezifischen Kommandos!

Vor der Überprüfung eines Dateisystems müssen Sie **umount** ausführen. Zur Überprüfung des **root**-Dateisystems führen Sie zuerst **touch /forcefsck** aus und starten dann den Rechner neu. Fehlerhafte Dateien bzw. Dateifragmente werden im Verzeichnis **lost+found** gespeichert.

- ▶ **-A**  
überprüft alle in **/etc/fstab** genannten Dateisysteme.
- ▶ **-t typ**  
gibt den Typ des Dateisystems an (etwa **ext3**, **xfs**).

```
fsck.ext2 [optionen] device
fsck.ext3 [optionen] device
fsck.ext4 [optionen] device
```

**fsck.ext2/3/4** bzw. **e2fsck** überprüft die Konsistenz eines **ext**-Dateisystems und führt gegebenenfalls Reparaturen durch. Das Kommando kann auch via **fsck** gestartet werden.

- ▶ **-b n**  
liest den alternativen Superblock **n**.
- ▶ **-n**  
beantwortet alle Rückfragen mit **n** (**no**), führt keine Änderungen durch.



- ▶ `-p`  
führt Reparaturen (Änderungen) im Dateisystem ohne Rückfrage durch.
- ▶ `-y`  
beantwortet alle Rückfragen mit `yes`, führt Änderungen durch.

```
fstrim [optionen] mount-verzeichnis
```

`fstrim` führt ein Trim-Kommando für das durch den `mount`-Punkt angegebene Dateisystem aus. Dabei werden standardmäßig alle nicht mehr vom Dateisystem genutzten Datenblöcke an den Datenträger gemeldet.

Das Kommando ist nur zweckmäßig, wenn es sich bei dem Datenträger um eine Solid State Disk (SSD) handelt und das Dateisystem nicht ohnedies bei jedem Löschvorgang automatisch Trim-Kommandos ausführt (siehe die `mount`-Option `discard` bei den Dateisystemen `ext4`, `btrfs` und `xfs`). `fstrim` kann täglich oder wöchentlich durch einen Cron-Job ausgeführt werden, um Performance-Verluste durch zu häufige oder aber gar nicht ausgeführte Trim-Kommandos zu vermeiden.

- ▶ `-l` bzw. `--length n`  
meldet ungenutzte Blöcke nur für den angegebenen Bereich.
- ▶ `-m` bzw. `--minimum n`  
meldet ungenutzte zusammenhängende Blöcke nur, wenn sie zumindest `n` Byte groß sind. Das kann die Ausführung des Kommandos erheblich beschleunigen.
- ▶ `-o` bzw. `--offset n`  
meldet ungenutzte Blöcke erst ab dem angegebenen Offset (gerechnet vom Beginn des Dateisystems). Die Angabe kann durch ein entsprechendes Suffix in binären `kByte`, `MByte`, `GByte` oder `TByte` (`K`, `M`, `G` oder `T`) oder in dezimalen `kByte`, `MByte` etc. angegeben werden (`KB`, `MB`, `GB` oder `TB`).
- ▶ `-v` bzw. `--verbose`  
liefert die Summe der mit dem Trim-Kommando gemeldeten Bytes.

### Beispiel

Das folgende Kommando meldet alle zumindest 64 KByte großen freien Datenblöcke an die SSD:

```
root# fstrim -m 64K /
/: 13493051392 bytes were trimmed
```

```
ftp [optionen] ftpserver
```

ftp stellt eine Verbindung zum angegebenen FTP-Server her. Nach dem Login können Sie interaktiv Dateien zwischen dem lokalen Rechner und dem FTP-Server übertragen. Im Folgenden sind die wichtigsten Kommandos für die interaktive Nutzung zusammengefasst:

- ▶ `?`  
zeigt eine Liste aller FTP-Kommandos an.
- ▶ `!`  
ermöglicht die Ausführung von Shell-Kommandos.
- ▶ `ascii`  
wechselt in den Textmodus.
- ▶ `binary`  
wechselt in den Binärmodus.
- ▶ `bye` oder `quit`  
beendet FTP.
- ▶ `cd verz`  
wechselt in das angegebene FTP-Verzeichnis.
- ▶ `get datei`  
überträgt die Datei vom FTP-Archiv in das aktuelle Verzeichnis.
- ▶ `lcd verz`  
wechselt das aktuelle Verzeichnis auf dem lokalen Rechner.
- ▶ `ls`  
zeigt die Liste der Dateien auf dem FTP-Server an.
- ▶ `lls`  
zeigt die Liste der Dateien auf dem lokalen Rechner an.
- ▶ `mget *.muster`  
überträgt alle passenden Dateien vom FTP-Archiv in das aktuelle Verzeichnis (siehe auch `prompt`).

- ▶ `open`  
stellt die Verbindung zum fremden Rechner her (wenn es beim ersten Versuch nicht geklappt hat).
- ▶ `prompt`  
aktiviert bzw. deaktiviert die Rückfrage vor der Übertragung jeder Datei durch `mget`.
- ▶ `put datei`  
überträgt die Datei vom aktuellen Verzeichnis in das FTP-Archiv (*upload*).
- ▶ `reget datei`  
setzt die Übertragung einer bereits teilweise übertragenen Datei fort.

```
[function] name
{kommandos}
```

Das optionale Schlüsselwort `function` definiert in `bash`-Scripts eine Subfunktion, die wie ein neues Kommando aufgerufen werden kann. Innerhalb der Funktion können mit `local` lokale Variablen definiert werden. Funktionen können rekursiv aufgerufen werden. Funktionen müssen *vor* ihrem ersten Aufruf deklariert werden!

An Funktionen können Parameter übergeben werden. Anders als bei vielen Programmiersprachen dürfen die Parameter nicht in Klammern gestellt werden. Innerhalb der Funktion können die Parameter den Variablen `$1`, `$2` etc. entnommen werden.

### Beispiel

Das folgende Mini-Script gibt *Hello World, abc!* aus:

```
#!/bin/bash
function myfunc {
    echo "Hello World, $1!"
}

myfunc "abc"
```

```
fuser dateiname
```

`fuser` ermittelt das Programm, das auf die angegebene Datei zugreift und diese möglicherweise für andere Programme blockiert. `fuser` liefert normalerweise nur die Prozessnummer (PID) des Programms.

► -k

beendet alle Prozesse, die auf die Datei zugreifen.

► -m

listet alle Prozesse auf, die auf *irgendeine* Datei des Dateisystems zugreifen, das als Device oder durch eine Datei spezifiziert ist.

► -v

gibt auch an, welcher Benutzer den Prozess ausführt (*verbose mode*).

Bei der Ausgabe folgt der Prozessnummer ein Buchstabe, der die Art des Zugriffs angibt:

► c

Der Prozess hat das Verzeichnis als aktuelles Verzeichnis eingestellt (*current directory*).

► e

Die Datei wird ausgeführt (*executable being run*).

► f

Die Datei ist für den Lesezugriff geöffnet (*open file*). Dieser Buchstabe wird nur im *verbose mode* angezeigt.

► F

Die Datei ist für den Schreibzugriff geöffnet. Auch F wird nur im *verbose mode* angezeigt.

► m

Die Datei wird als Bibliothek oder durch Memory-Mapping genutzt.

► r

Das Verzeichnis wird als Wurzelverzeichnis genutzt (*root directory*).

Eine Alternative zu `fuser` ist das Kommando `lsof`.

### Beispiel

Der Versuch, eine DVD durch `umount` aus dem Dateisystem zu lösen, scheitert. Schuld ist der Prozess 32664 des Benutzers `kofler`, der mit `less` gerade eine Textdatei der DVD liest.

```

root# mount /run/media/kofler/myBackupDvd
mount: /run/media/kofler/myBackupDvd: target is busy.
root# fuser -v -m /run/media/kofler/myBackupDvd
                BEN.          PID ZUGR.  BEFEHL
/run/media/kofler/myBackupDvd
      kofler      32664 f.... less

```

### gconftool-2 Kommando

Das Gnome-2-Kommando `gconftool-2` liest Werte aus der `gconf`-Datenbank bzw. speichert Einstellungen dort. Die `gconf`-Datenbank besteht aus zahlreichen kleinen XML-Dateien im Verzeichnis `.gconf`. Die Datenbank wird von älteren Gnome-Programmen zur Speicherung diverser Einstellungen verwendet.

- ▶ `-get pfad`  
liefert den Wert für den Parameter, der durch den Pfad angegebenen wird. Der Pfad ergibt sich aus dem Verzeichnis und dem Schlüsselnamen.
- ▶ `-set pfad -type datentyp wert`  
speichert einen neuen Wert. Zulässige Datentypen sind `int`, `bool`, `float` und `string`.

Als Alternative zu `gconftool-2` können Sie auch den grafischen Einstellungs-Editor `gconf-editor` verwenden. Die meisten Gnome-3-Programme speichern ihre Einstellungen hingegen in der binären `dconf`-Datenbank. Diese kann mit dem Kommando `gsettings` oder mit der grafischen Benutzeroberfläche `dconf-editor` verändert werden.

### gdisk [optionen] [device]

`gdisk` ist ein interaktives Programm zur Partitionierung von Festplatten mit einer GUID-Partitionstabelle (GPT). `gdisk` ist das Gegenstück zum Kommando `fdisk` für MBR-Partitionstabellen. Die Bedienung von `gdisk` weist daher viele Ähnlichkeiten zu `fdisk` auf. Alle in Tabelle 11 beschriebenen Tastaturkürzel gelten auch für `gdisk`.

Beachten Sie, dass `gdisk` nach einer Rückfrage eventuell existierende MBR-Partitionstabellen durch eine neue GUID-Partitionstabelle ersetzt. `gdisk` ist weit weniger populär als `fdisk`. Als Standardwerkzeug zur Partitionierung von GPT-Datenträgern gilt `parted` oder dessen grafische Benutzeroberfläche `gparted`.

```
genisoimage [optionen] verzeichnis1 verzeichnis2
```

`genisoimage` (ehemals `mkisofs`) erzeugt ein ISO-Image, das anschließend auf eine CD oder DVD gebrannt werden kann, z. B. mit `wodim`. Grundsätzlich werden an `genisoimage` alle Verzeichnisse übergeben, deren Inhalt in das Wurzelverzeichnis des ISO-9660-Dateisystems eingefügt werden soll. Die Verzeichnisse selbst werden nicht Teil des ISO-Dateisystems! Wenn Sie das möchten, müssen Sie die Option `-graft-points` verwenden und die Verzeichnisse in der Form `verz1=verz1 verz2=verz2` angeben.

`genisoimage` schreibt die resultierende ISO-Datei in die Standardausgabe. Oft wird die Ausgabe mit `-o dateiname` in eine Datei umgeleitet. Daneben gibt es zahllose weitere Optionen, von denen hier nur die wichtigsten erwähnt sind.

► `-apple`

nutzt die Apple-Extension, sodass die resultierende CD auch auf Apple-Rechnern gelesen werden kann. (Alternativ können Sie auch die Option `-hfs` einsetzen – siehe unten.)

► `-b datei`

nutzt die angegebene Datei als Boot-Image. Beim Rechnerstart wird diese Datei automatisch ausgeführt (sofern das BIOS entsprechend konfiguriert ist). Normalerweise erwartet `genisoimage` ein Disketten-Image (1,2, 1,44 oder 2,88 MByte). Wenn die Bootdatei eine andere Größe bzw. ein anderes Format hat, müssen Sie zusätzlich die Option `-hard-disk-boot` oder `-no-emul-boot` angeben. Die Dateiangabe erfolgt relativ zum Dateisystem der zu brennenden CD (nicht relativ zum aktuellen Dateisystem auf der Festplatte!).

► `-boot-info-table`

integriert einen 56 Byte langen Informationsblock (die sogenannte *El Torito boot info table*) in die Boot-Image-Datei. Die Option ist nur in Kombination mit `-b` sinnvoll.

Vorsicht: Dadurch wird die Boot-Image-Datei auf der Festplatte verändert! Erstellen Sie gegebenenfalls vorher eine Sicherheitskopie.

► `-boot-load-size n`

gibt an, wie viele 512-Byte-Sektoren des Boot-Images gelesen werden sollen. Die Option ist nur in Kombination mit `-b` und `-no-emul-boot` zweckmäßig. Normalerweise wird dann die gesamte Bootdatei gelesen. Manche BIOS-Versionen verlangen, dass `n` ein Vielfaches von 4 ist.

► `-c name`

gibt an, unter welchem Dateinamen der sogenannte Bootkatalog auf der CD gespeichert werden soll. Der Katalog wird von `genisoimage` erzeugt. Sie müssen bei der Angabe des Namens lediglich darauf achten, dass es keinen Konflikt mit einer vorhandenen Datei gibt. (Der `man`-Text rät dazu, den Namen `boot.catalog` zu verwenden.)

► `-D`

erlaubt mehr als acht Verzeichnisebenen. Das widerspricht dem ISO-Standard und kann auf manchen Systemen, die eine exakte Einhaltung des ISO-Standards verlangen, zu Problemen führen. Ohne diese Option hält sich `genisoimage` an den Standard und verändert gegebenenfalls die Verzeichnishierarchie und die Namen der betroffenen Dateien.

► `-dvd-video`

berücksichtigt die Regeln für Video-DVDs (UDF-Unterstützung, korrekte Sortierung der Dateien etc.). Achten Sie darauf, dass die Dateinamen der Video-Dateien in Großbuchstaben angegeben werden müssen!

► `-f`

verfolgt symbolische Links. Der Inhalt der so angegebenen Dateien oder Unterverzeichnisse wird in das ISO-Image mit aufgenommen. (Ohne diese Option wird nur der Link an sich im ISO-Image gespeichert. Symbolische Links können allerdings nur auf CDs mit Rockridge-Extension abgebildet werden.)

► `-graft-points`

erlaubt Verzeichnisangaben in der Form `iso=real`. Das bedeutet, dass das Verzeichnis `real` im ISO-Image den Namen `iso` bekommt. Sie können mit diesem Mechanismus auch den Pfad von Verzeichnissen ändern oder einzelne Dateien umbenennen. Beispielsweise erreichen Sie mit der folgenden Option, dass der Inhalt des Verzeichnisses `/data/fotos/2014/2014-01-diverse` auf der CD oder DVD im Verzeichnis `2014-01` gespeichert wird:

```
-graft-points 2014-01=/data/fotos/2014/2014-01-diverse
```

Oft wird die Option nur deswegen verwendet, um zu erreichen, dass das zu speichernde Verzeichnis an sich (und nicht nur sein Inhalt) in das ISO-Image aufgenommen wird. In diesem Fall stimmen der ISO-Name und der reale Name überein (`verz1=verz1 verz2=verz2`).

► `-hard-disk-boot`

gibt an, dass das mit `-b` angegebene Boot-Image ein Festplatten-Image ist. Die Image-Datei muss mit einem MBR (*Master Boot Record*) beginnen und eine Partition enthalten.

► `-hfs`

erzeugt eine Hybrid-CD, die sowohl das ISO-9660- als auch das unter Mac OS X übliche HFS-Dateisystem verwendet. Die eigentlichen Dateien werden aber nur einmal gespeichert. Durch diverse weitere Optionen können Sie diverse HFS-Details einstellen. Eine umfassende Beschreibung aller HFS-Optionen gibt `genisoimage`.

► `-input-charset name`

gibt an, welchen Zeichensatz die Dateinamen im vorhandenen Dateisystem nutzen. Normalerweise verändert `genisoimage` die Zeichensatzcodierung für das Standard-ISO-Format und für die Rockridge-Extension nicht. Eine Liste der möglichen Zeichensätze liefert `genisoimage -input-charset help`.

► `-J`

nutzt die Joliet-Extension (Microsoft), um Informationen über lange Dateinamen zu speichern. Dateinamen werden in Unicode (UTF-16) codiert. Die Option kann auch in Kombination mit `-R` verwendet werden – dann ist das Ergebnis eine Hybrid-CD, bei der lange Dateinamen sowohl unter Unix als auch unter Windows sichtbar sind.

► `-joliet-long`

erlaubt bis zu 103 Zeichen in Dateinamen. (Die Joliet-Extension beschränkt Dateinamen normalerweise auf 64 Zeichen.)

► `-nobak`

ignoriert alle Dateien, die die Zeichen `#` oder `~` enthalten oder mit `.bak` enden.

► `-no-emul-boot`

gibt an, dass das Boot-Image (Option `-b`) ein sogenanntes *no emulation image* ist. Die Datei wird beim Booten direkt von der CD/DVD gelesen, ohne dabei eine Diskette oder eine Festplatte zu emulieren.

► `-o datei`

speichert das ISO-Image in der angegebenen Datei.



► `-output-charset name`

gibt an, welchen Zeichensatz die Dateinamen im ISO-Image nutzen sollen. Die Einstellung gilt nur für das Standard-ISO-Format und die Rockridge-Extension. Normalerweise belässt `genisoimage` die Zeichencodierung so, wie sie ist. Die Option hat keine Konsequenzen für die Joliet-Extension – dort kommt immer Unicode zum Einsatz.

► `-R`

nutzt die Rockridge-Extension (Unix), um Informationen über lange Dateinamen zu speichern. Die Option kann auch in Kombination mit `-J` verwendet werden – dann ist das Ergebnis eine Hybrid-CD.

► `-r`

hat dieselbe Wirkung wie `-R`, setzt aber für alle Dateien die UID und GID auf 0 (entspricht `root`). Gleichzeitig werden alle R-Zugriffsbits gesetzt, außerdem alle X-Zugriffsbits, wenn in der ursprünglichen Datei zumindest ein X-Bit gesetzt war (siehe auch die Beschreibung des `chmod`-Kommandos). Das Ergebnis: Jeder darf alle Dateien und Verzeichnisse lesen und Programme ausführen. Wenn Sie die Zugriffsrechte exakter steuern möchten, können Sie dies mit den Optionen `-uid`, `-gid`, `-dir-mode`, `-file-mode` und `-new-file-mode tun` (siehe man `genisoimage`).

► `-udf`

ergänzt das ISO-Image um UDF-Verwaltungsinformationen.

► `-V name`

gibt den Namen der CD/DVD an (die sogenannte »Volume-ID«).

## Beispiele

Das folgende Kommando schreibt alle Dateien innerhalb des `data`-Verzeichnisses in die Datei `/tmp/master.iso`. Das Verzeichnis `data` ist selbst *kein* Verzeichnis im ISO-Image. Das ISO-Image nutzt sowohl die Rockridge-Extension (Option `-r`) als auch die Joliet-Extension (Option `-J`) und bekommt den Namen `Linux` (Option `-V`). Wenn Sie aus dem ISO-Image eine CD brennen, gilt diese Zeichenkette als CD-Name.

```
user$ genisoimage -o /tmp/master.iso -r -J -V Linux data
```

Das zweite Beispiel ist dem ersten ähnlich, allerdings wird diesmal eine bootfähige CD erstellt:

```
user$ genisoimage -o /tmp/master.iso -r -J data -b images/boot.img \
      -c boot.catalog
```

Im dritten Beispiel ist das Verzeichnis `data` nun selbst ein Verzeichnis im ISO-Image (Option `-graft-points`):

```
user$ genisoimage -o /tmp/master.iso -r -graft-points /data=/data
```

**getcap** [optionen] dateiname

`getcap` aus dem Paket `libcap` oder `libcap-progs` ermittelt bei ausführbaren Dateien, welche Operationen für das Programm zulässig sind (Capabilities). Obwohl der Linux-Kernel bereits seit 2009 Capabilities unterstützt, wird diese Funktion momentan nur sehr sporadisch genutzt.

Capabilities setzen ein Dateisystem mit Extended Attributes voraus. Bei ext-Dateisystemen muss die `mount`-Option `user_xattr` verwendet werden.

### Beispiel

Zu den wenigen Kommandos, bei denen Capabilities bei einigen Distributionen genutzt werden, zählt `ping`. Damit dieses Kommando von gewöhnlichen Benutzern verwendet werden kann, hat das Kommando die Erlaubnis, `cap_net_raw+ep` Operationen auszuführen, also elementare Netzwerkfunktionen. Das folgende `getcap`-Ergebnis kam unter openSUSE Leap zustande. Sie sehen, dass von allen Kommandos in `/usr/bin` außer `ping` in seinen IPv4- und IPv6-Varianten nur das Kommando `cdrecord` Capabilities nutzt:

```
root# getcap /usr/bin/*
/usr/bin/cdrecord = cap_ipc_lock,cap_sys_rawio,cap_sys_nice,cap_sys_resource+ep
/usr/bin/ping     = cap_net_raw+ep
/usr/bin/ping6    = cap_net_raw+ep
```

Wie Sie Capabilities selbst einstellen, zeigt ein Beispiel bei der Beschreibung des setcap-Kommandos.

**getenforce**

`getenforce` liefert den Zustand des SELinux-Systems. Die möglichen Ergebnisse sind Enforcing, Permissive oder Disabled. Zur Veränderung des Zustands verwenden Sie das Kommando setenforce.

```
getfacl [optionen] dateiname
```

getfacl ermittelt die erweiterten Zugriffsrechte der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem ACLs (*Access Control Lists*) unterstützt. Bei ext3/ext4-Dateisystemen muss dazu die mount-Option `acl` verwendet werden. Ein Beispiel finden Sie bei der Beschreibung des Kommandos setfacl.

- ▶ `-d`  
zeigt die Standard-ACLs an.
- ▶ `-R`  
zeigt die ACLs aller Dateien in allen Unterverzeichnissen an.
- ▶ `--skip-base`  
liefert keine Ergebnisse bei Dateien, für die nur die gewöhnlichen Unix-Zugriffsrechte gelten, aber keine ACL-Regeln.
- ▶ `--tabular`  
zeigt die ACLs in Tabellenform an.

```
getfattr [optionen] dateiname
```

getfattr ermittelt die erweiterten Attribute der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem *Extended Attributes* unterstützt (mount-Option `user_xattr` bei ext-Dateisystemen). Ein Beispiel finden Sie bei der Beschreibung des Kommandos setfattr.

- ▶ `-d`  
liefert eine Liste aller Benutzerattribute und ihrer Einstellungen. Als Standardmuster für den Attributnamen gilt »user«.
- ▶ `-m attributmuster`  
liefert die Attribute, deren Namen dem angegebenen Muster entsprechen.
- ▶ `-n attributname`  
liefert den Wert des angegebenen Attributs. Es muss der vollständige Attributname angegeben werden, beispielsweise `user.attrname`.
- ▶ `-R`  
zeigt die EAs aller Dateien in allen Unterverzeichnissen an.

```
git kommando [optionen/parameter]
```

Das Kommando `git` aus dem gleichnamigen Paket steuert das Versionsverwaltungssystem Git. Es wurde ursprünglich nur zur Kernelentwicklung konzipiert, zählt heute aber neben Subversion zu den am weitesten verbreiteten Versionsverwaltungssystemen. Anders als bei Subversion hat jeder Git-Nutzer ein eigenes, lokales Git-Repository, die sogenannte *working copy*. Auch Repository-Zweige (*branches*) sind lokal!

Dateien in Git-Repositories können drei Zustände annehmen:

- ▶ *committed*: Die lokale Datei ist im Repository gespeichert und wurde seither nicht mehr verändert.
- ▶ *modified*: Die lokale Datei wurde verändert, die Veränderungen wurden aber noch nicht durch einen Commit im Repository gespeichert.
- ▶ *staged*: Eine veränderte Datei (*modified*) soll beim nächsten Commit berücksichtigt werden.

### Kommandos

- ▶ `add dateien/verzeichnisse`

fügt Dateien oder Verzeichnisse dem Repository hinzu bzw. kennzeichnet bereits im Repository befindliche, lokal veränderte Dateien für den nächsten Commit. Damit ändert sich der Status der Dateien von *modified* zu *staged*.

- ▶ `branch [-v]`

listet die lokalen Repository-Zweige auf und markiert den aktuellen Zweig mit einem vorangestellten Stern. Mit der Option `-v` zeigt `git` zusätzlich den letzten Commit in jedem Branch an.

- ▶ `branch zweigname`

erzeugt einen neuen Repository-Zweig. Um in diesen Zweig zu wechseln, müssen Sie außerdem `git checkout zweigname` ausführen! Bis zu diesem Zeitpunkt gelten alle Commits für den Default-Zweig `master` bzw. für den zuletzt eingestellten Zweig.

Repository-Zweige sind in Git nicht einfach Verzeichnisse. Zweige dienen vielmehr dazu, den Kontext für Commits bzw. die daraus resultierenden Snapshots festzulegen.

► `branch -d zweigname`

löscht den angegebenen Zweig. Das Kommando wird nur ausgeführt, wenn der Zweig vorher mit einem anderen Zweig oder mit HEAD verbunden wurde. Um einen Zweig unabhängig vom Merge-Status zu löschen, verwenden Sie die Option `-D`.

► `branch -u reponame/zweigname`

richtet eine Verknüpfung (in der Git-Nomenklatur: einen *track*) zwischen dem gerade aktuellen lokalen Zweig und dem externen Zweig `reponame/zweigname` ein. Wenn es nur ein externes Git-Repository gibt, ersetzen Sie `reponame` durch `origin`. Im Regelfall werden der lokale und der externe Zweigname übereinstimmen, das ist aber nicht zwingend erforderlich. Die Verknüpfung bewirkt, dass die betreffenden lokalen Zweige durch `git pull` und `git push` automatisch verbunden werden, und erspart Ihnen so manuelle `merge`-Kommandos.

► `checkout [-b] zweigname`

wechselt in den angegebenen Zweig. Damit werden alle Dateien im Arbeitsverzeichnis in den Zustand versetzt, den sie beim letzten Commit in diesem Zweig hatten. Mit der Option `-b` wird der angegebene Branch zuerst erzeugt und dann aktiviert. `git checkout -b b1` entspricht also `git branch b1; git checkout b1`.

► `clone repo-url [verzeichnis]`

lädt ein externes Git-Repository herunter und erstellt eine lokale Kopie im aktuellen bzw. im angegebenen Verzeichnis. Git-URLs haben z.B. die Form `git://host.com/sample/sample.git` oder `ssh://host.com/sample/sample.git`. Das externe Repository wird zukünftig unter dem Namen `origin` angesprochen. Git erlaubt die Verwendung mehrerer externer Repositories, die mit `git remote` verwaltet werden.

► `commit [-a] -m 'snapshot beschreibung'`

erstellt einen neuen, lokalen Snapshot im Git-Repository. Darin werden alle Änderungen seit dem letzten Commit gespeichert, aber nur, sofern der Status der Dateien *staged* ist. Sie müssen also vorher alle zu speichernden Dateien mit `git add` als *staged* kennzeichnen.

Mit der zusätzlichen Option `-a` können Sie sich diese Mühe sparen – `git commit` berücksichtigt nun *alle* Änderungen und führt für die betroffenen Dateien automatisch `git add` aus. Damit verhält sich `git commit -a` ähnlich wie `svn commit`.

Der letzte Snapshot hat immer den Namen `HEAD`. Der vorletzte Snapshot kann mit `HEAD^` angesprochen werden, der vorvorletzte mit `HEAD^2` etc. Vor dem ersten Commit müssen Sie Ihren Namen und Ihre E-Mail-Adresse mit `git config` in der Git-Konfiguration speichern.

► `config [--global] parameter wert`

speichert die Einstellung eines Git-Parameters in der Datei `.git/config` der lokalen Git-Repositories. Mit `--global` wird der Parameter in der globalen Git-Konfigurationsdatei `.gitconfig` gespeichert und gilt somit als Default für mehrere Repositories.

► `diff`

ohne weitere Parameter zeigt alle seit dem letzten Commit durchgeführten Änderungen, die nicht für den nächsten Commit gekennzeichnet sind. `git diff` zeigt also nur die Änderungen von Dateien im Status *modified*, berücksichtigt aber keine Dateien im Status *staged*.

► `diff --cached`

zeigt alle *staged*-Änderungen – also alle Änderungen, die beim nächsten Commit als Snapshot im Git-Repository gespeichert werden.

► `diff HEAD`

zeigt alle Änderungen seit dem letzten Commit, sowohl *modified* als auch *staged*.

► `fetch [reponame]`

lädt die zuletzt durchgeführten Änderungen von einem externen Git-Repository herunter. Standardmäßig wird dazu das bei `clone` angegebene Repository ausgelesen, also `origin`. `git fetch` aktualisiert also das lokale Repository und integriert alle neuen externen Commits, Zweige und Tags. Um Namenskonflikte zu vermeiden, bekommen neue externe Zweige den Namen `origin/zweigname` bzw. `reponame/zweigname`.

Zu `git fetch` gibt es die Variante `git pull`: Damit werden die externen Änderungen nicht nur heruntergeladen, sondern auch gleich mit dem lokalen Zweig verbunden. `git pull` entspricht `git fetch`; `git merge`. Dabei verbindet `merge` lokale Zweige mit den verfolgten Remote-Zweigen (*tracked remote branch*, siehe auch `git branch -u`).

► `init`

richtet für das aktuelle Verzeichnis ein Git-Repository ein.

► `log [--oneline] [--graph] [zweig/tag [^nichtzweig]]`

liefert eine Liste der zuletzt durchgeführten Commits. Die Option `--oneline` verkürzt die Ausgabe pro Commit auf eine Zeile. `--graph` visualisiert, wie Zweige miteinander verbunden wurden.

Wenn Sie als optionalen Parameter einen Zweig oder einen Markierungspunkt (Tag) angeben, erscheinen nur solche Commits, die Einfluss auf diesen Zweig hatten. Dabei kann es sich durchaus auch um Commits handeln, die *vor* dem Anlegen dieses Zweigs erfolgten, weil diese den Ausgangspunkt des Zweiges formten. Um die von `git log zweigname` produzierte Commit-Liste weiter zu reduzieren, können Sie mit `^nichtzweig` Commits des angegebenen Zweigs ausblenden. `git log bugfix ^master` liefert somit nur solche Commits des `bugfix`-Zweigs, die noch *nicht* mit `master` verbunden wurden.

Die Ausgabe wird durch `less` geleitet, sodass Sie mit den Cursortasten durch das zumeist seitenlange Ergebnis blättern können. Dieses Verhalten können Sie bei allen `git`-Kommandos durch die Option `--no-pager` unterbinden.

► `merge zweigname`

verbindet den angegebenen Zweig mit dem gerade aktuellen Zweig. Die sich daraus ergebenden Änderungen werden wie bei `git commit` in einem neuen Snapshot gespeichert. Um extern durchgeführte Änderungen zu verbinden, laden Sie diese zuerst herunter (`git fetch`) und führen dann `git merge reponame/zweigname` bzw. `git merge origin/zweigname` aus.

► `pull [reponame]`

entspricht `git fetch`; `git merge`. Die Details sind beim `fetch`-Kommando beschrieben. Häufig ist es zweckmäßiger, `fetch` und `merge` getrennt auszuführen, um Missverständnisse zu vermeiden, welcher Zweig womit verbunden wird.

► `push [-u] [reponame [zweigname]]`

lädt lokal durchgeführte Commits für `zweigname` in das externe Git-Repository `reponame` hoch und verbindet sie dort (`merge`). Beispielsweise verbindet `git push origin master` alle eigenen Commits im `master`-Zweig mit dem gleichnamigen Zweig des externen Repositories `origin`. Wenn `zweigname` im externen Repository nicht existiert, wird der Zweig dort automatisch eingerichtet. Durch die Option `-u` werden der lokale Zweig und der gleichnamige externe Zweig verknüpft (wie durch `git branch -u`).

Das `push`-Kommando scheitert, wenn es externe Commits gibt, die lokal noch nicht durchgeführt sind. Für das vorige Beispiel müssten Sie vorher `git fetch` und `git merge origin/master` ausführen.

Wenn Sie `git push` ausführen, ohne Repository- und Zweignamen anzugeben, kommuniziert `git` mit dem Default-Repository (zumeist `origin`) und synchronisiert den aktuellen verknüpften Zweig (siehe das Kommando `branch -u`).

► `remote [-v]`

listet alle externen Repositories auf. Mit `-v` zeigt `git` auch die URL jedes Repositories an. Das Stamm-Repository `origin` wird durch `git clone` definiert. Weitere Repositories können mit Kommandos wie `git remote add` oder `git remote remove` verwaltet werden.

► `reset HEAD -- datei`

ändert den Status einer irrtümlich als *staged* gekennzeichneten Datei zurück zu *modified*.

► `reset [--hard] xxx`

macht Änderungen rückgängig und führt zurück in eine alte Version des Repository. `xxx` gibt die gewünschte Version an: `HEAD` führt zurück zum letzten Commit, `HEAD^` zum vorletzten, `HEAD^2` zum vorvorletzten etc. Alternativ können Sie auch die Anfangsziffern der hexadezimalen SHA1-Prüfsumme des Commits angeben. `reset origin/master` widerruft alle lokalen Änderungen und synchronisiert den lokalen Zweig mit dem `master`-Zweig des Default-Repository. Die Option `--hard` bewirkt, dass lokale Änderungen unwiderruflich überschrieben werden.

► `rm datei`

löscht die Datei aus dem lokalen Verzeichnis und entfernt sie aus dem *staging*-Bereich.

► `status [-s]`

liefert Informationen über den aktuellen Git-Zweig sowie über den Status veränderter Dateien. Die Option `-s` macht die Statusausgabe kompakter. Allen Dateinamen werden zwei Buchstaben vorangestellt, wobei der erste Buchstabe den *staging*- und der zweite den *modifying*-Zustand angibt.

► `tag -a 'marker-text' [-m 'kommentar'] [commitid]`

fügt dem letzten Commit ein *tag* hinzu, also eine Markierung, eine Art Lesezeichen. Diese Funktion eignet sich gut, um z. B. eine Release-Version zu kennzeichnen. Wenn Sie nachträglich einen älteren Commit markieren möchten, müssen Sie dessen ID angeben. Das ist eine achtstellige hexadezimale Zahl, die Sie z. B. mit `git log` ermitteln können. Wird `git tag` ohne weitere Parameter ausgeführt, liefert das Kommando eine alphabetische Liste aller Tags.

Zu `git` gibt es nicht eine *man*-Seite, sondern eine ganze Menge: Jedes `git`-Kommando ist in einer eigenen Seite dokumentiert. Beispielsweise gibt man `git-clone` eine umfassende Beschreibung zu `git clone`.



## Beispiel

Das Beispiel zeigt, wie Sie einen schnellen Bugfix im `master`-Zweig durchführen. Die ersten drei `git`-Kommandos stellen sicher, dass Sie den lokalen `master`-Zweig verwenden und dass dieser aktuell ist. `git commit` erzeugt einen neuen lokalen Snapshot für alle geänderten Dateien. `git push` lädt die durchgeführten Änderungen in das `origin`-Repository hoch und verbindet sie mit dessen `master`-Zweig.

```
user$ git checkout master
user$ git fetch
user$ git merge origin/master master
user$ meinlieblingseditor dateien ...
user$ git commit -a -m 'Bugfix Login-Authentifizierung'
user$ git push origin master
```

## glxinfo [optionen]

Mit dem Programm `glxinfo`, das sich je nach Distribution im Paket `mesa-utils`, `glx-utils` oder `Mesa-demo-x` versteckt, können Sie die 3D-Funktionen Ihres Systems überprüfen. Das Programm liefert eine Menge Detailinformationen über das laufende GLX-System, also über die OpenGL-3D-Erweiterungen des X Window Systems. Aus den sehr detaillierten Ergebnissen filtern Sie die relevanten Zeilen am besten mit `grep` heraus.

## Beispiel

Mit `glxinfo` können Sie herausfinden, welcher Grafiktreiber auf Ihrem Rechner läuft:

```
user$ glxinfo | grep -i vendor      (in einer virtuellen Maschine)
server glx vendor string: SGI
client glx vendor string: Mesa Projet and SGI
OpenGL vendor string: VMware, Inc.
user$ glxinfo | grep -i vendor      (NVIDIA-Grafikkarte, proprietärer NVIDIA-Treiber)
server glx vendor string: NVIDIA Corporation
client glx vendor string: NVIDIA Corporation
OpenGL vendor string: NVIDIA Corporation
user$ glxinfo | grep -i vendor      (Intel-Chipset, Open-Source-Treiber)
server glx vendor string: SGI
client glx vendor string: Mesa Projet and SGI
OpenGL vendor string: Intel Open Source Technology Center
```

Wenn Sie speziell an den 3D-Funktionen interessiert sind, werfen Sie einen Blick auf Zeilen, die die Zeichenkette `render` enthalten. Die folgende Ausgabe ist auf einem Rechner mit ATI-Grafikkarte entstanden. Der verwendete Treiber stellt 3D-Funktionen zur Verfügung:

```
root# glxinfo | grep render
direct rendering: Yes
OpenGL renderer string: ATI Radeon R9 M290X OpenGL Engine
```

Wenn dagegen kein 3D-beschleunigter Treiber läuft, sieht die Ausgabe wie in einem der drei folgenden Beispiele aus. Im ersten Beispiel stehen gar keine 3D-Funktionen zur Verfügung, im zweiten Fall werden sie per Software durch die Mesa-Bibliothek nachgebildet und im dritten Fall durch die llvmpipe-Bibliothek. Diese Bibliothek erhöht zwar den Rechenaufwand der CPU beträchtlich, ermöglicht aber bei vielen Distributionen die Nutzung von 3D-Funktionen ohne einen richtigen 3D-Grafiktreiber.

```
root# glxinfo | grep render
Xlib: extension "GLX" missing on display ":0.0"
root# glxinfo | grep render
direct rendering: No
OpenGL renderer string: Mesa GLX Indirect
root# glxinfo | grep render
direct rendering: Yes
OpenGL renderer string: Gallium 0.4 on llvmpipe (LLVM 3.6, 256 bits)
```

### **gnome-session-quit** [optionen]

Das Programm `gnome-session-quit` zeigt einen Dialog an, in dem Sie sich abmelden bzw. den Rechner ausschalten oder neu starten können.

- ▶ `--force`  
führt den Logout auch dann durch, wenn es noch ungespeicherte Dateien gibt. Diese Option kann nur in Kombination mit `--logout` verwendet werden.
- ▶ `--logout`  
fragt den Benutzer, ob er sich abmelden möchte.
- ▶ `--no-prompt`  
führt den Logout ohne Rückfrage aus. Diese Option kann nur in Kombination mit `--logout` verwendet werden.
- ▶ `--power-off`  
fragt den Benutzer, ob er den Rechner ausschalten möchte.
- ▶ `--reboot`  
fragt den Benutzer, ob er den Rechner neu starten möchte.

**gpg[2]** [optionen/kommandos]

gpg verschlüsselt, entschlüsselt oder signiert Dateien und verwaltet Schlüssel. gpg wird durch unzählige Optionen gesteuert und ist für den manuellen Einsatz nur bedingt geeignet. Das Kommando wird unter anderem von E-Mail-Clients zur Verschlüsselung und Signierung von Nachrichten sowie von Paketverwaltungswerkzeugen zur Verifizierung der Integrität von Paketen und Paketquellen eingesetzt.

Mit den Optionen `-c` und `-d` kann gpg zur symmetrischen Ver- und Entschlüsselung von Dateien verwendet werden. Das ist einfach, aber sicherheitstechnisch nicht optimal. Alle anderen Optionen gehen davon aus, dass Sie eine asymmetrische Verschlüsselung verwenden möchten: Die Verschlüsselung erfolgt dann mit dem öffentlichen Teil des Schlüssels Ihres Kommunikationspartners, die Entschlüsselung mit dem privaten Teil Ihres eigenen Schlüssels. Das setzt voraus, dass Sie gpg auch zur Schlüsselverwaltung verwenden.

gpg ist für den Server- und Embedded-Einsatz gedacht und greift auf weniger externe Bibliotheken zurück. gpg2 ist hingegen für den Desktop-Einsatz konzipiert. Diese Version unterstützt zusätzlich S/MIME. Davon abgesehen unterstützen beide Kommandos dieselben Optionen.

► `--batch`

aktiviert den Batch-Modus. In diesem sind interaktive Kommandos nicht zulässig, es erfolgt keine Interaktion. Der Batch-Modus ist zur Script-Programmierung gedacht.

► `-c datei` bzw. `--symmetric datei`

fordert zur Eingabe eines Passworts auf, verschlüsselt dann die angegebene Datei und speichert das Ergebnis unter dem Namen `datei.gpg`.

► `--cipher-algo zip/zlib/bzip2/none`

legt den Verschlüsselungsalgorithmus fest. Standardmäßig verwendet gpg den Algorithmus CAST5. Die zur Auswahl stehenden Algorithmen ermittelt das Kommando `gpg --version`.

► `--compress-algo zip/zlib/bzip2/none`

gibt an, ob bzw. wie die Daten komprimiert werden sollen. Standardmäßig wird das ZIP-Verfahren eingesetzt. Aus Kompatibilitätsgründen werden andere Einstellungen als `zip` oder `none` nicht empfohlen. `none` bietet sich dann an, wenn die Ausgangsdaten bereits komprimiert sind. Die neuerliche Komprimierung kostet dann unnötig Zeit, ohne den Platzbedarf zu reduzieren.

- ▶ `-d datei.gpg` bzw. `--decrypt datei.gpg`  
fordert zur Eingabe eines Passworts auf, entschlüsselt dann die angegebene Datei und schreibt das Ergebnis in die Standardausgabe.
- ▶ `--decrypt datei > klartext`  
entschlüsselt die angegebene Datei und leitet die Ausgabe in die Datei `klartext` um. Die Datei muss mit dem öffentlichen Teil Ihres eigenen Schlüssels verschlüsselt sein. Zum Entschlüsseln müssen Sie das Passwort Ihres Schlüssels angeben.
- ▶ `--encrypt datei`  
verschlüsselt die angegebene Datei und speichert sie als `datei.gpg`. Die E-Mail-Adresse des Schlüssels geben Sie interaktiv an. Der Schlüssel muss gegebenenfalls zuvor mit `--import` in der Schlüsseldatenbank gespeichert werden. Wenn Sie vor der Option `--encrypt` zusätzlich `--armor` angeben, wird die verschlüsselte Datei im ASCII-Format gespeichert. Das erfordert mehr Platz, vereinfacht aber die Weitergabe per E-Mail.
- ▶ `--export name@host > public.bin`  
speichert eine Binärdarstellung des öffentlichen Teils des Schlüssels für `name@host` in der angegebenen Datei. Diese Datei müssen Sie an Personen weitergeben, die Ihnen verschlüsselte Dateien oder E-Mails senden sollen. Wenn Sie vor der Option `--export` zusätzlich `--armor` angeben, wird der öffentliche Schlüssel in einer E-Mail-tauglichen Textdarstellung gespeichert.
- ▶ `--gen-key`  
erzeugt einen neuen Schlüssel und speichert diesen in `.gnupg/pubring.gpg`. Dabei müssen Sie diverse Daten angeben – die gewünschte Art und Länge des Schlüssels, Ihren Namen, die E-Mail-Adresse, ein Passwort etc.
- ▶ `--import public-key-file`  
importiert den öffentlichen Teil eines Schlüssels einer anderen Person in die eigene Schlüsselsammlung. Das ist notwendig, um dieser Person verschlüsselte Dateien bzw. E-Mails senden zu können.
- ▶ `-k` bzw. `--list-keys`  
listet alle in `.gnupg/pubring.gpg` gespeicherten Schlüssel auf.
- ▶ `-no-tty`  
unterdrückt die Ausgabe von Warnungen.
- ▶ `--passphrase-file filename`  
liest das Verschlüsselungspasswort aus der angegebenen Datei.

- ▶ `--passphrase-repeat n`  
gibt an, wie oft die Passwordeingabe wiederholt werden muss (standardmäßig einmal). Wenn das Passwort in Scripts aus einer Datei gelesen wird (siehe die obige Option), deaktivieren Sie die wiederholte Eingabe mit dem Parameter 0.
- ▶ `-q` bzw. `--quiet`  
verzichtet (weitgehend) auf Status-Ausgaben.
- ▶ `--version`  
zeigt die Version von gpg und liefert eine Liste aller unterstützten Verschlüsselungs- und Hash-Algorithmen.

## Beispiel

Das folgende Kommando führt eine symmetrische Verschlüsselung des Dokuments geheim.odt durch:

```
user gpg -c geheim.odt
Geben Sie die Passphrase ein: *****
Geben Sie die Passphrase nochmals ein: *****
```

gpg erzeugt damit die neue Datei geheim.odt.gpg. Die ursprüngliche Datei geheim.odt wird nicht verändert. Wenn Sie den Zugriff auf diese Datei im lokalen Dateisystem verhindern möchten, müssen Sie die Datei sowie deren Backups explizit löschen! Um die verschlüsselte Datei wiederherzustellen, gehen Sie so vor:

```
user gpg -d geheim.odt.gpg > geheim.odt
Geben Sie die Passphrase ein: *****
```

Im zweiten Beispiel wird gpg in zwei Scripts zum Ver- und Entschlüsseln von Dateien mit dem AES256-Algorithmus eingesetzt. Voraussetzung für die Scripts ist eine Schlüsseldatei im Textmodus, die mit pwgen erzeugt wird:

```
root# pwgen -1 -y 80 > key.txt
root# chmod 400 key.txt
```

Das Verschlüsselungs-Script mycrypt sieht so aus:

```
#!/bin/sh
# Verwendung: mycrypt < in > out.crypt
gpg -c --batch --compress-algo none --passphrase-file key.txt \
  --passphrase-repeat 0 --cipher-algo AES256
```

Um die verschlüsselte Datei wieder zu dechiffrieren, verwenden Sie myuncrypt:

```
#!/bin/sh
# Verwendung: myuncrypt < in.crypt > out
gpg -d --batch --no-tty -q --compress-algo none --passphrase-file key.txt \
  --passphrase-repeat 0 --cipher-algo AES256
```

Die Anwendung sieht so aus:

```
root# ./mycrypt < backup.tar.gz > backup.tar.gz.crypt
root# ./myuncrypt < backup.tar.gz.crypt > backup-copy.tar.gz
root# file backup.tar.gz.crypt
backup.tar.gz.crypt: GPG symmetrically encrypted data (AES256 cipher)
root# diff backup.tar.gz backup-copy.tar.gz
(keine Ausgabe, die Dateien sind identisch)
```

**gpio** [optionen] kommando

Das Kommando `gpio` des WiringPi-Projekts (<http://wiringpi.com>) steht standardmäßig unter Raspbian zur Verfügung und wird dazu verwendet, um den Zustand der GPIO-Pins des Raspberry Pi auszulesen bzw. zu verändern. `gpio` kann auch einige weitere Hardware-Funktionen des Raspberry Pi steuern.

`gpio` unterstützt drei verschiedene Nomenklaturen zur Bezeichnung der Pins. Normalerweise verwendet das Kommando eine eigene Nomenklatur, die Sie auf der Projektwebseite bzw. mit `gpio readall` ergründen können (Spalte `wPi`). Wenn Sie die Pins anders ansprechen möchten, müssen Sie das Kommando mit den Optionen `-1` oder `-g` aufrufen.

- ▶ `-1` (eins, nicht L!)  
verwendet die Pin-Nummern des J8-Headers (1-40).
- ▶ `-g`  
verwendet die GPIO-Nummerierung gemäß der Dokumentation von Broadcom für den Chip BCM2835/2836.
- ▶ `-v`  
zeigt die Versionsnummer von `gpio` sowie Details zum Raspberry-Pi-Modell, auf dem das Kommando ausgeführt wird.

Alle weiteren Operationen werden durch Subkommandos ausgeführt:

- ▶ `mode n in|out|pwm|clock|up|down|tri`  
gibt an, in welchem Modus der Pin `n` verwendet werden soll. Dabei ist zu beachten, dass die Pins des Raspberry Pi zwar unterschiedliche Modi unterstützen, aber nicht jeder Pin unterstützt jeden Modus. Beispielweise können Sie nur den Ausgang »GPIO 18« (Pin 12) zur Hardware-Pulsweitenmodulation (`pwm`) nutzen.
- ▶ `pwm n 0-1023`  
stellt den PWM-Duty-Cycle im Wertebereich zwischen 0 und 1023 ein.

- ▶ `read n`  
liest den Zustand des Signaleingangs von Pin `n`.
- ▶ `readall`  
liefert eine Tabelle aller GPIO-Pins mit Informationen über deren aktuellen Zustand. Dabei bezeichnet die Spalte `Physical` die Pin-Nummern des Headers, die Spalte `BCM` die Pin-Nummern gemäß der BCM-Dokumentation und `wPi` die Nummern gemäß der WiringPi-Nomenklatur.
- ▶ `reset`  
aktiviert für alle GPIO-Pins den Input-Modus.
- ▶ `wfi n rising|falling|wait`  
wartet, bis sich der Zustand von Pin `n` ändert.
- ▶ `write n 0/1`  
setzt den Signalausgang von Pin `n` auf 0 V (Low) bzw. 3,3 V (High).

## Beispiele

Das folgende Beispiel adressiert Pins gemäß der Nummerierung des J8-Headers. Pin 26 soll als Ausgang verwendet werden, z. B. um damit eine Leuchtdiode ein- und auszuschaalten.

```
pi$ gpio -1 mode 26 out    (Pin 26 als Ausgang verwenden)
pi$ gpio -1 write 26 1     (Ausgang auf High stellen)
pi$ gpio -1 write 26 0     (Ausgang auf Low stellen)
```

Im zweiten Beispiel wird Pin 12 verwendet, um mit PWM die Helligkeit einer Leuchtdiode einzustellen:

```
pi$ gpio -1 mode 12 pwm    (Pin 12 als PWM-Ausgang verwenden)
pi$ gpio -1 pwm 12 0       (LED aus)
pi$ gpio -1 pwm 12 500     (LED in mittlerer Helligkeit)
pi$ gpio -1 pwm 12 1023    (LED in maximaler Helligkeit)
```

```
grep [optionen] suchmuster datei
grep -R [optionen] suchmuster
```

`grep` durchsucht die angegebene Textdatei nach einem Suchmuster und zeigt die gefundenen Textpassagen an oder gibt einfach nur an, wo bzw. in wie vielen Zeilen das Muster gefunden wurde. `grep` kann mit `find` kombiniert werden, um alle Dateien, die bestimmten Bedingungen entsprechen, nach Texten zu durchsuchen.

- ▶ `-n`

zeigt nicht nur die Zeile mit dem gefundenen Text an, sondern zusätzlich die *n* unmittelbar vorangehenden und nachfolgenden Zeilen.
- ▶ `-c`

gibt nur die Anzahl der Zeilen an, in denen das Suchmuster gefunden wurde, nicht aber die Zeilen selbst.
- ▶ `--color=auto`

kennzeichnet die Übereinstimmungen bei der Ausgabe farbig. Das funktioniert nur, wenn Sie das `grep`-Ergebnis direkt in einer Konsole anzeigen, nicht aber, wenn Sie das Ergebnis mit einem anderen Kommando weiterverarbeiten (z. B. mit `sort` oder `less`).
- ▶ `-E`

aktiviert *Extended Regular Expressions* (EREs). Diese erweiterte Syntax erlaubt die Zeichen `?` (null oder einmal), `+` (mindestens einmal) und `|` (logisches Oder) im Suchmuster.
- ▶ `-f Dateiname`

liest die hier aufgezählten Optionen der angegebenen Datei (für komplexe oder häufig benötigte Suchmuster).
- ▶ `-i`

unterscheidet nicht zwischen Groß- und Kleinschreibung.
- ▶ `-l`

zeigt nur die Dateinamen an, in denen das Suchmuster gefunden wurde.
- ▶ `-n`

gibt bei der Ausgabe jeder Zeile auch deren Zeilennummer an.
- ▶ `-P`

interpretiert das reguläre Muster gemäß Perl-Regeln.
- ▶ `-q`

liefert keine Bildschirmausgaben und gibt lediglich den Rückgabewert 0 (Suchtext gefunden) oder 1 (nicht gefunden) zurück. Diese Option ist sinnvoll, wenn `grep` von anderen Programmen aufgerufen wird (siehe das Beispiel bei [find](#)).
- ▶ `-R`

wendet `grep` rekursiv auf alle Dateien des Verzeichnisbaums an.



## ► -v

inverse Wirkung: grep liefert alle Zeilen, die dem Suchmuster *nicht* entsprechen.  
 grep -v '^#' datei | cat -s liefert alle Zeilen der Datei, die nicht mit # beginnen  
 (wobei cat -s mehrere Leerzeilen auf eine einzige reduziert).

## ► -w

findet nur ganze Wörter. Wenn diese Option angegeben ist, wird das Suchmuster  
 »die« im Wort »dieser« nicht mehr erkannt.

Das Suchmuster besteht aus zwei Komponenten: aus der Angabe, wonach gesucht wird, und aus der Angabe, wie oft der Suchausdruck auftreten darf. Tabelle 1 fasst die Bedeutungen der wichtigsten Zeichen zusammen. Das Kürzel ERE markiert dabei Zeichen für die *Extended Regular Expressions* (Option -E oder Kommando egrep).

Wenn Sonderzeichen wie ? \* + [ ] ( ) # oder ! im Suchmuster verwendet werden sollen, muss \ vorangestellt werden. Für manche Zeichengruppen sind bereits Muster vordefiniert, etwa [:digit:] für Ziffern oder [:space:] für *White Space* (also Leer- und Tabulatorzeichen, siehe man-Seite).

Zeichen	Bedeutung
abc	die Zeichenkette »abc«
[abc]	eines der Zeichen a, b oder c
[^abc]	keines der Zeichen a, b oder c (also ein beliebiges anderes Zeichen)
[a-z]	eines der Zeichen zwischen a und z
.	ein beliebiges Zeichen
?	Das vorangegangene Zeichen darf gar nicht oder einmal auftreten (ERE).
*	Das Zeichen darf beliebig oft (auch gar nicht) auftreten.
+	Das Zeichen darf beliebig oft, muss aber mindestens einmal auftreten (ERE).
{n}	Das Zeichen muss genau n-mal auftreten.
{,n}	Das Zeichen darf höchstens n-mal auftreten.
{n,}	Das Zeichen muss mindestens n-mal auftreten.
{n,m}	Das Zeichen muss mindestens n- und höchstens m-mal auftreten.
^	Zeilenanfang
\$	Zeilenende
	Logisches Oder (ERE)

**Tabelle 1** Aufbau von grep-Suchmustern

Einige Beispiele für Suchmuster: 'abc' sucht nach genau dieser Zeichenkette. '[a-z][0-9]+' sucht nach Zeichenketten, die mit einem Kleinbuchstaben beginnen und in denen eine oder mehrere Ziffern folgen. '\(.\*\) ' sucht nach beliebigen Zeichenketten, die in zwei runde Klammern eingeschlossen sind.

grep wendet das Suchmuster nur zeilenweise an. Textpassagen, die durch einen Zeilenumbruch unterbrochen sind, können nicht erkannt werden.

### Beispiele

Das folgende Kommando sucht aus der Apache-Logging-Datei /var/log/apache2/access.log alle Zeilen heraus, die die IP-Adresse 1.2.3.4 enthalten. Das Kommando muss mit root-Rechten ausgeführt werden, weil gewöhnliche Benutzer kein Zugriffsrecht auf die Apache-Logging-Dateien haben.

```
root# grep 1.2.3.4 /var/log/apache2/access.log
```

Das folgende Kommando ermittelt, wie oft die Zeichenkette mysql in beliebiger Groß- und Kleinschreibung in einer PHP-Datei verwendet wurde:

```
user$ grep -c -i mysql code.php
```

Im nächsten Beispiel werden ausgehend von einem Startverzeichnis rekursiv alle Apache-Konfigurationsdateien durchsucht. grep zeigt die Dateinamen an, in denen das Schlüsselwort ErrorDocument vorkommt:

```
root# cd /etc/apache2
root# grep -R -l ErrorDocument
```

Das folgende grep-Kommando filtert aus den Kernelnachrichten alle Zeilen heraus, die in beliebiger Groß- und Kleinschreibung das Suchmuster eth enthalten:

```
root# dmesg | grep -i eth
e1000: eth0 NIC Link is Down
e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
...
```

Die Ausgabe des Kommandos df enthält diverse Einträge zu temporären Dateisystemen, die oft uninteressant sind. grep -v eliminiert diese Zeilen:

```
root# df | grep -v tmpfs
Dateisystem      1K-Blöcke Benutzt Verfügbar Verw% Eingehängt auf
/dev/mapper/fedora-root 16080304 4820060 10420360 32% /
/dev/sda1         487652 109669 348287 24% /boot
```

Manche Konfigurationsdateien im Verzeichnis /etc sehen so aus, als hätten die Autoren die Datei mit dem Handbuch verwechselt. Zwischen Hunderten, bisweilen Tausenden von Kommentarzeilen sind wenige tatsächlich wirksame Einstellungen versteckt. Die folgenden Kommandos eliminieren alle Zeilen, die mit # oder ; beginnen bzw. die leer sind:

```
root# cd /etc/samba
root# cp smb.conf smb.conf.orig
root# grep -Ev '^#|^;$' smb.conf.orig > smb.conf
```

Um auch Kommentare zu eliminieren, bei denen # oder ; nicht am Zeilenanfang steht, sowie leere Zeilen mit Whitespace (Leerzeichen, Tabulator-Zeichen), formulieren Sie das grep-Kommando so:

```
root# grep -Ev '^[[:space:]]*#|^[[:space:]]*^;|^^[[:space:]]*$' \
    smb.conf.orig > smb.conf
```

Sie können `find` und `grep` auch kombinieren, um besonders wirkungsvolle Suchen durchzuführen. Im folgenden Kommando sucht `find` nach \*.php-Dateien und `grep` überprüft, ob in ihnen die Zeichenkette `mysql` vorkommt. Wenn das der Fall ist, wird der Dateiname auf dem Bildschirm ausgegeben. Beachten Sie, dass die Option `-print` nach `-exec` angegeben werden muss.

```
root# find -name '*.php' -exec grep -q -i mysql {} \; -print
```

### Varianten zu grep

- ▶ `egrep` entspricht `grep -E` und unterstützt eine erweiterte Syntax für das Suchmuster. Beispielsweise liefert `egrep 'muster1|muster2' datei` alle Zeilen der angegebenen Datei, die eines der beiden Muster enthält.
- ▶ `fgrep` entspricht `grep -F` und liest die Suchmuster zeilenweise aus einer angegebenen Datei.
- ▶ `grepmail` hilft dabei, E-Mail-Archive im mbox-Format nach Zeichenketten zu durchsuchen. Der Vorteil gegenüber `grep` besteht darin, dass nicht eine einzelne Zeile, sondern die gesamte E-Mail aus der Datei extrahiert wird.
- ▶ `zgrep` durchsucht komprimierte Dateien.

### groupadd name

`groupadd` richtet eine neue Gruppe ein.

- ▶ `-g n`  
verwendet `n` als neue GID-Nummer (Group Identification).

### groupdel name

`groupdel` löscht die angegebene Gruppe.

```
groupmod [optionen] name
```

groupmod verändert die GID-Nummer und/oder den Gruppennamen der Gruppe.

- ▶ `-g n`  
bestimmt die neue GID-Nummer (Group Identification).
- ▶ `-n name`  
bestimmt den neuen Gruppennamen.

### Beispiel

Das folgende Kommando gibt der Gruppe `mysqladmins` den neuen Namen `dbadmins`:

```
root# groupmod -n dbadmins mysqladmins
```

```
groups [benutzername]
```

groups zeigt eine Liste aller Gruppen an, denen der aktuelle bzw. der angegebene Benutzer angehört. Dabei handelt es sich um die in `/etc/passwd` angegebene Hauptgruppe sowie um die optional in `/etc/group` angegebenen Gruppen.

### Beispiel

Der aktuelle Benutzer gehört zu den Gruppen `kofler`, `docuteam` und `wheel`. Mitglieder der `wheel`-Gruppe dürfen in RHEL 6 administrative Aufgaben erledigen.

```
user$ groups  
kofler docuteam wheel
```

```
growisofs [optionen] verzeichnis1 verzeichnis2 ...  
growisofs [optionen] -Z device=iso-datei
```

growisofs speichert die angegebenen Verzeichnisse bzw. die ISO-Datei auf einer DVD oder einer Blu-ray-Disc. Das Kommando greift auf `genisoimage` zurück und nutzt bis auf wenige Ausnahmen dieselben Optionen. Es gibt nur wenige zusätzliche Optionen:

- ▶ `-dry-run`  
führt einen Probelauf durch, ohne tatsächlich Daten zu schreiben.
- ▶ `-dvd-compat`  
erhöht die Kompatibilität zu DVD-ROM-Laufwerken. DVD-Rs bzw. DVD+Rs werden abgeschlossen (und können daher nicht durch weitere Sessions erweitert

werden). Bei DVD+RWs wird nach den Daten der (an sich optionale) *Lead-out*-Bereich geschrieben.

► *-M device*

gibt ebenfalls das DVD-Device an. Bei Multi-Session-DVDs verwenden Sie bei der ersten Session *-Z*, bei allen weiteren Sessions *-M*. *growisofs* kümmert sich selbstständig darum, dass *genisoimage* die Session an der korrekten Position beginnt.

Achten Sie darauf, dass Sie bei Multi-Sessions immer dieselbe Kombination von Optionen angeben, die einen Einfluss auf das ISO-Dateisystem haben (Joliet- und Rockridge-Erweiterungen etc.).

► *-speed=n*

gibt die gewünschte Schreibgeschwindigkeit an. Das ist normalerweise nicht notwendig – das Laufwerk bestimmt selbst die korrekte Geschwindigkeit. Sie können aber durch *-speed=1* ein langsames Brennen erzwingen.

► *-Z device*

gibt das DVD-Device an.

## Beispiele

Das folgende Kommando speichert den Inhalt des Verzeichnisses *data* auf einer DVD. Die Optionen *-r* und *-J* bewirken, dass die DVD lange Dateinamen entsprechend den Rockridge- und Joliet-Erweiterungen aufweist.

```
user$ growisofs -r -J -Z /dev/sr0 data/
```

Eine zweite Session fügen Sie so hinzu (Option *-M* statt *-Z*):

```
user$ growisofs -r -J -M /dev/sr0 moredata/
```

Dieses Kommando schreibt den Inhalt der ISO-Datei *daten.iso* auf die DVD:

```
user$ growisofs -Z /dev/sr0=daten.iso
```

## grub

Das Kommando *grub* steht bei Distributionen zur Verfügung, die die alte GRUB-Version 0.97 als Bootloader verwenden. Zu den wenigen Distributionen mit GRUB 0.97, die noch eine nennenswerte Verbreitung haben, zählen CentOS 6 bzw. RHEL 6.

*grub* startet eine interaktive GRUB-Shell. Darin führen Sie Kommandos zur Installation von GRUB aus. Alternativ können Sie GRUB auch in der Form *grub < commandfile* starten, um alle in *commandfile* enthaltenen GRUB-Kommandos auszuführen.

Beim Start von `grub` können Sie diverse Optionen übergeben, von denen hier aber nur die allerwichtigsten aufgezählt sind:

- ▶ `--config-file=name`  
gibt den Namen der GRUB-Konfigurationsdatei an (bei CentOS und RHEL `/boot/grub/grub.conf`, sonst oft `/boot/grub/menu.lst`).
- ▶ `--no-floppy`  
verzichtet auf die Suche nach einem Diskettenlaufwerk.
- ▶ `--read-only`  
startet GRUB im Testmodus. GRUB führt keinerlei Schreiboperationen aus.

### Beispiel

Die folgenden Kommandos schreiben GRUB in den MBR der ersten Festplatte. Die Bootpartition mit der GRUB-Konfigurationsdatei ist `/dev/sda2`, also `(hd0,1)` in der GRUB-Nomenklatur. Ich setze hier voraus, dass die GRUB-Konfigurationsdatei `/boot/grub/menu.lst` vorher korrekt eingerichtet wurde.

```
root# grub
grub> root (hd0,1)
grub> setup (hd0)
grub> quit
```

```
grub-install [device]
grub2-install [device]
```

Das Kommando `grub-install` (Ubuntu, Debian) bzw. `grub2-install` (CentOS, Fedora, SUSE, RHEL) steht bei allen Distributionen zur Verfügung, die GRUB 2 als Bootloader verwenden. `grub-install` installiert den Bootloader in den Bootsektor des angegebenen Festplatten-Devices bzw. bei EFI-Rechnern in die EFI-Partition. `grub-install` setzt voraus, dass zuvor die GRUB-Konfigurationsdatei `/boot/grub/grub.cfg` eingerichtet wurde, in der Regel durch den Aufruf von `update-grub` (Debian, Ubuntu) oder mit `grub2-mkconfig` (CentOS, Fedora, SUSE, RHEL).

### Beispiel

Unter Ubuntu erstellt das Kommando `update-grub` die GRUB-Konfigurationsdatei `/boot/grub/grub.cfg`, wobei automatisch Einträge für alle auf dem Rechner gefundenen Betriebssysteme angelegt werden. `grub-install` installiert GRUB 2 bei BIOS-Rechnern in den Bootsektor der ersten Festplatte:

```
root# update-grub
root# grub-install /dev/sda
```

Bei einem EFI-Rechner entfällt die Device-Angabe. `grub-install` setzt voraus, dass die EFI-Partition im Verzeichnis `/boot/efi` in den Verzeichnisbaum eingebunden ist.

```
root# update-grub
root# grub-install /dev/sda
```

Unter CentOS, Fedora, SUSE und RHEL müssen Sie stattdessen diese Kommandos ausführen, mit oder ohne Device-Angabe bei `grub2-install`, je nachdem, ob es sich um eine BIOS- oder um eine EFI-Installation handelt:

```
root# grub2-mkconfig -o /boot/grub2/grub.cfg
root# grub2-install [/dev/sda]
```

```
grub-mkconfig [optionen]
grub2-mkconfig [optionen]
```

`grub-mkconfig` (Ubuntu, Debian) bzw. `grub2-mkconfig` (CentOS, Fedora, SUSE, RHEL) steht bei Distributionen zur Verfügung, die GRUB 2 als Bootloader verwenden. Das Kommando wertet die GRUB-Konfigurationsdateien in `/etc/grub.d` aus und analysiert alle Festplattenpartitionen. Aus diesen Informationen erzeugt das Kommando eine neue GRUB-Konfigurationsdatei und gibt diese aus.

► `-o datei`

speichert die resultierende Konfiguration in die angegebene Datei.

```
gsettings kommando
```

`gsettings` liest Werte aus der `dconf`-Datenbank bzw. speichert Einstellungen dort. Die `dconf`-Datenbank befindet sich üblicherweise in der Datei `.config/dconf/user` und liegt in einem Binärformat vor. Die Datenbank wird zur Speicherung diverser Einstellungen von neueren Gnome-Programmen verwendet.

► `get schema schlüssel`

ermittelt den Wert des Parameters `schlüssel` für das angegebene Schema. Das »Schema« enthält den Software-Anbieter, den Programmnamen und (optional) ein Einstellungsverzeichnis. Beispielsweise sind alle Nautilus-Einstellungen für die Icon-Ansicht dem Schema `org.gnome.nautilus.icon-view` zugeordnet.

- ▶ `list-keys schema`  
liefert eine Liste aller Schlüssel für das angegebene Schema.
- ▶ `list-schemas`  
liefert eine nicht sortierte Liste aller Schemata.
- ▶ `set schema schlüssel wert`  
weist dem Parameter schlüssel einen neuen Wert zu.

### Beispiel

Das folgende Kommando stellt den Textskalierungsfaktor ein. Damit werden alle Texte um 66 Prozent vergrößert dargestellt. Das ist z. B. bei einem HiDPI- bzw. Retina-Monitor zweckmäßig, auf dem die Texte sonst unleserlich klein sind.

```
user$ gsettings set org.gnome.desktop.interface text-scaling-factor 1.66
```

**gtf** x y freq

gtf berechnet eine Generalized Timing Formula, also eine Sequenz von Zahlen, die einen VESA-Grafikmodus beschreibt. Die Zahlensequenz kann zur Definition eines Grafikmodus in der Konfigurationsdatei /etc/X11/xorg.conf bzw. durch das Kommando xrandr verwendet werden.

### Beispiel

Das folgende Kommando berechnet die Parameter für den Grafikmodus 1280 × 720 bei einer Bildfrequenz von 60 Hertz:

```
user$ gtf 1280 720 60
# 1280x720 @ 60.00 Hz (GTF) hsync: 44.76 kHz; pclk: 74.48 MHz
Modeline "1280x720_60.00" 74.48 1280 1336 1472 1664 720 721 724 746 \
-HSync +Vsync
```

**gunzip** datei

Dekomprimiert die angegebene Datei, unabhängig davon, ob sie mit gzip oder mit compress komprimiert wurde. Dabei wird automatisch die Kennung .gz bzw. .Z im Dateinamen entfernt. gunzip ist ein Link auf gzip, wobei automatisch die Option -d aktiviert ist.



**gzip** [optionen] datei

gzip komprimiert bzw. dekomprimiert die angegebene Datei. Komprimierten Dateien wird automatisch die Kennung `.gz` angehängt. gzip eignet sich nur zur Komprimierung einzelner Dateien. Wenn Sie mehrere Dateien (oder ganze Verzeichnisse) in einer komprimierten Datei speichern möchten, müssen Sie zusätzlich das Kommando `tar` verwenden.

► `-1` bis `-9`, `--fast`, `--best`

steuert die Geschwindigkeit und Qualität der Kompression. `-1` entspricht `--fast` und bewirkt eine schnelle Kompression, aber etwas größere Dateien. `-9` entspricht `--best` und führt zu höheren Rechenzeiten, aber kleineren Dateien. Die Grundeinstellung ist `-6`.

► `-c` bzw. `--stdout` oder `--to-stdout`

lässt die zu (de)komprimierende Datei unverändert und leitet das Ergebnis auf die Standardausgabe (in der Regel auf den Bildschirm) um. Von dort kann es mit `>` in eine beliebige Datei umgeleitet werden (siehe Beispiel unten).

► `-d` bzw. `--decompress` oder `--uncompress`

dekomprimiert die angegebene Datei, anstatt sie zu komprimieren (entspricht `gunzip`).

► `-r` bzw. `--recursive`

(de)komprimiert auch Dateien in allen Unterverzeichnissen.

## Beispiele

Das folgende Kommando komprimiert alle `*.tif`-Dateien des aktuellen Verzeichnisses. Das Ergebnis sind lauter `*.tif.gz`-Dateien.

```
user$ gzip *.tif
```

Im zweiten Beispiel komprimiert gzip die angegebene Datei, lässt sie aber unverändert und schreibt das Ergebnis in `backup.gz`:

```
user$ gzip -c datei > backup.gz
```

**halt** [optionen]

halt beendet alle laufenden Prozesse und fährt den Rechner dann herunter. halt entspricht `shutdown -h now`.

► `-p`

schaltet den Rechner nach dem Shutdown aus (*poweroff*). Bei vielen Systemen erfolgt das Ausschalten automatisch. Die Option ist nur erforderlich, wenn dies nicht funktioniert – z. B. in vielen virtuellen Maschinen.

### `hash [option]`

Das `bash`-Kommando `hash` zeigt den Inhalt der Hash-Tabelle an. Dabei handelt es sich um eine Tabelle, in der sich die Shell die Pfadnamen aller bereits ausgeführten Kommandos merkt. Dadurch wird die abermalige Ausführung eines bereits bekannten Kommandos beschleunigt, weil jetzt nicht mehr alle `PATH`-Verzeichnisse nach dem Programm durchsucht werden müssen.

► `-r`

löscht die Hash-Tabelle der `bash`. Das ist notwendig, wenn sich das Verzeichnis eines Programms ändert, das sich in der Hash-Tabelle befindet. Die `bash` findet das Kommando sonst nicht mehr. In der `tcsh` muss statt `hash -r` das Kommando `rehash` verwendet werden.

### `hciconfig [hcidvice] [kommando]`

`hciconfig` aus dem Paket `bluez` hilft bei der Konfiguration lokaler Bluetooth-Adapter über das Host Controller Interface (HCI). Sofern kein HCI-Device-Name angegeben wird (üblicherweise `hci0` oder `hci1`), kommuniziert `hciconfig` mit allen lokalen Bluetooth-Geräten. Wenn das Kommando ohne Parameter ausgeführt wird, listet es Informationen über alle lokalen Bluetooth-Adapter auf.

► `reset`

initiiert einen Neustart des Bluetooth-Adapters.

► `up/down`

aktiviert bzw. deaktiviert den Bluetooth-Adapter. Wenn dabei die Fehlermeldung *Operation not possible due to RF-kill* angezeigt wird, können Sie versuchen, den Adapter vorher mit `rfkill unblock` einzuschalten.

```
hcidtool [optionen] [kommando]
```

Das Kommando `hcidtool` aus dem Paket `bluez` hilft beim Scannen und Einrichten von Bluetooth-Geräten.

- ▶ `-h`  
listet alle unterstützten Kommandos auf.
- ▶ `-i hciX`  
wendet das folgende Kommando auf das angegebene Bluetooth-Device an. Ohne diese Option wird das Kommando an das erste verfügbare Bluetooth-Gerät gesendet.

### Kommandos

- ▶ `cc bt-mac`  
stellt eine Verbindung zum Bluetooth-Gerät mit der angegebenen MAC-Adresse her.
- ▶ `dc bt-mac`  
beendet die angegebene Verbindung wieder.
- ▶ `dev`  
liefert eine Liste der lokalen Bluetooth-Geräte. Normalerweise handelt es sich dabei um den eingebauten Bluetooth-Adapter, dem in der Regel der Device-Name `hci0` zugewiesen ist.
- ▶ `scan`  
listet alle in Funkreichweite befindlichen externen Bluetooth-Geräte inklusive ihrer MAC-Adressen auf.

### Beispiel

Es gibt nur einen lokalen Bluetooth-Adapter mit dem Device-Namen `hci0`. Alle weiteren Kommandos werden daher automatisch an diesen Adapter gesendet; die Option `-i` kann entfallen. In Funkreichweite befinden sich unter anderem ein Android-Smartphone und eine Maus.

```
user$ hcidtool dev
Devices:
      hci0    00:1F:CF:41:00:A2
```

```
user$ hcitool scan
60:FB:42:FC:BB:8C   Michael Koflers Maus
10:68:3F:25:68:18   Nexus 4
...
```

**hdparm** [optionen] device

hdparm liest bzw. verändert Parameter von IDE-Festplatten und IDE-CD/DVD-Laufwerken. Das Kommando eignet sich mit Einschränkungen auch zur Konfiguration von SATA- und SCSI-Festplatten.

- ▶ -d  
ermittelt, ob der DMA-Modus aktiv ist.
- ▶ -d1 bzw. -d0  
aktiviert bzw. deaktiviert den DMA-Modus.
- ▶ -i  
liefert die wichtigsten Daten und Einstellungen des Laufwerks. Die Daten wurden bereits während des Systemstarts gelesen und werden in einer kompakten Kurzform dargestellt.
- ▶ -I  
liefert detaillierte Informationen zum Laufwerk. Die Daten werden bei der Ausführung des Kommandos vom Laufwerk abgefragt und sind daher aktueller als bei hdparm -i.
- ▶ -S *n*  
stellt ein, nach welcher Zeit ohne Aktivität die Festplatte abgeschaltet wird (Spin-Down). Die Option ist vor allem bei Notebooks interessant, um Energie zu sparen. Die Interpretation von *n* ist recht merkwürdig:
  - 0 deaktiviert diese Funktion, d. h., die Festplatte wird nie ausgeschaltet.
  - 1 bis 240 entsprechen  $n \times 5$  Sekunden (also 5 Sekunden bis 20 Minuten).
  - 241 bis 251 entsprechen  $(n-240) \times 30$  Minuten (also 30 Minuten bis fünfeinhalb Stunden).
  - 252 entspricht einer Spin-Down-Zeit von 21 Minuten.
  - 253 aktiviert die herstellerspezifische Spin-Down-Zeit, die zwischen 8 und 12 Stunden liegt.

- 254 ist reserviert.
- 255 entspricht einer Spin-Down-Zeit von 21 Minuten und 15 Sekunden.

► **-t und -T**

ermittelt die Datenübertragungsrate von der Festplatte zum Rechner, wobei die Daten bei **-t** tatsächlich vom Datenträger, bei **-T** hingegen aus dem Cache-Speicher gelesen werden.

► **-W1 bzw. -W0**

aktiviert bzw. deaktiviert den Write-Cache der Festplatte.

**head** [optionen] datei

**head** gibt die ersten zehn Zeilen einer Textdatei auf dem Bildschirm aus.

► **-n zeilen**

gibt die angegebene Anzahl von Zeilen aus.

**help** name

**help** zeigt eine kurze Beschreibung des angegebenen **bash**-Kommandos an. **help** funktioniert nur für Kommandos, die in die **bash** integriert sind, beispielsweise **alias**, **cd** oder **type**.

**host** [optionen] name/ip-adresse

**host** liefert die IP-Adresse zum angegebenen Netzwerknamen bzw. den Netzwerknamen zur angegebenen IP-Adresse. Bei vielen Distributionen ist **host** ein Bestandteil eines **bind**-Pakets, z. B. **bind-utils** oder **bind9-host**.

► **-a**

liefert zusätzliche Informationen, die eventuell helfen, Fehler in der Nameserver-Konfiguration zu finden.

► **-t typ**

liefert DNS-Einträge des gewünschten Typs.

## Beispiel

Dem Hostnamen `kofler.info` sind die IP-Adressen `5.9.22.28` und `2a01:4f8:161:107::3` zugewiesen. Der Mail-Server hat den Domainnamen `mail.kofler.info`. Die SPF-Informationen des Mail-Servers gehen aus dem DNS-Texteintrag hervor.

```
user$ host kofler.info
kofler.info has address 5.9.22.29
kofler.info has IPv6 address 2a01:4f8:161:107::4
kofler.info mail is handled by 10 mail.kofler.info.
user$ host 5.9.22.29
28.22.9.5.in-addr.arpa domain name pointer kofler.info.
user$ host 2a01:4f8:161:107::4
4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.f.4.0.1.0.a.2.ip6.arpa
domain name pointer kofler.info.
user$ host -t txt kofler.info
kofler.info descriptive text "v=spf1 ip4:5.9.22.29 ip6:2a01:4f8:161:107::4 -all"
```

## hostname [name]

`hostname` liefert den aktuellen Netzwerknamen des Systems bzw. verändert ihn bis zum nächsten Neustart. Die Veränderung wird also nicht bleibend gespeichert! Wenn Sie das möchten, führen Sie `hostnamectl` aus oder verändern die entsprechende Konfigurationsdatei.

- ▶ **-d**  
liefert den Domainnamen anstelle des Hostnamens.
- ▶ **-i**  
liefert die dem Hostnamen zugeordnete IP-Adresse. Da es möglich ist, dass dem Host mehrere IP-Adressen zugeordnet sind, sollten Sie statt dieser Option nach Möglichkeit die Option **-I** verwenden.
- ▶ **-I**  
liefert alle dem Hostnamen zugeordneten IP-Adressen in einer durch Leerzeichen getrennten Liste. Die `localhost`-Adresse sowie lokale IPv6-Adressen werden dabei nicht berücksichtigt. Die Option **-I** steht bei älteren Versionen von `hostname` nicht zur Verfügung.

```
hostnamectl [optionen] [kommando]
```

Mit dem Kommando `hostnamectl` kann bei Distributionen mit dem Init-System `Systemd` der Hostname neu eingestellt werden. Anders als mit dem `hostname`-Kommando wird die neue Einstellung bleibend gespeichert.

► `set-hostname name`

stellt den neuen Hostnamen ein. Damit die Änderung wirksam wird, müssen Sie sich aus- und neu einloggen. Der neue Hostname wird gleichzeitig in `/etc/hostname` gespeichert.

► `status`

zeigt den aktuellen Hostnamen sowie diverse Angaben über das laufende System an, darunter den Distributionsnamen und die Kernelversion.

### Beispiel

Das folgende Kommando stellt den Hostnamen `arbeitsplatz` und den Domainnamen `mylan` ein:

```
root# hostnamectl set-hostname arbeitsplatz.mylan
```

```
htop [optionen]
```

`htop` aus dem gleichnamigen Paket ist eine Variante zu `top`. Der Hauptvorteil gegenüber `top` besteht darin, dass Sie mit den Cursortasten horizontal und vertikal durch die Prozessliste scrollen können. Der ausgewählte Prozess kann unkompliziert mit `[K]` beendet werden. Mit den Funktionstasten können Sie außerdem zwischen verschiedenen Darstellungsformen wählen.

► `-d n`

gibt an, nach wie vielen zehntel Sekunden die Prozessliste aktualisiert werden soll.

► `-p n1,n2,n3,...`

zeigt nur die Prozesse mit den angegebenen PIDs an.

► `-s name`

sortiert die Prozessliste nach dem angegebenen Kriterium. Eine Liste aller erlaubten Schlüsselwörter gibt `htop -s help`.

► `-u name`

zeigt nur Prozesse des ausgewählten Benutzers.

```
htpasswd [optionen] passwortdatei benutzername [passwort]
```

htpasswd bzw. bei manchen Distributionen htpasswd2 erzeugt eine Passwortdatei für den Apache-Webserver oder ändert Einträge in einer bereits vorhandenen Passwortdatei.

Die Datei kann zur Basic-Authentifizierung verwendet werden (AuthType Basic). Ihr Ort muss in einer Apache-Konfigurationsdatei (z.B. httpd.conf oder .htaccess) mit dem Schlüsselwort AuthUserFile angegeben werden. Aus Sicherheitsgründen sollten Sie unbedingt darauf achten, dass der Webserver die Datei zwar lesen kann, aber nicht via HTTP ausliefern darf!

- ▶ -b  
erwartet das Passwort als Parameter. Das vereinfacht die Erzeugung von Password-einträgen per Script, ist aber unsicher.
- ▶ -c  
erzeugt eine neue Passwortdatei.
- ▶ -D  
löscht einen Benutzer aus der Passwortdatei.
- ▶ -l  
sperrt den Account vorübergehend.
- ▶ -u  
aktiviert einen gesperrten Account wieder.

### Beispiel

Mit den folgenden Kommandos wird die neue Passwortdatei passwords.pwd erzeugt und dort ein Eintrag für den Benutzer name1 eingefügt. Weitere Benutzernamen/Passwort-Paare werden ohne die Option -c hinzugefügt:

```
user$ htpasswd -c passwords.pwd name1
New password: ******
user$ htpasswd passwords.pwd name2
New password: ******
```

```
iconv -f Zeichensatz1 -t Zeichensatz2 in.txt > out.txt
```

iconv führt eine Zeichensatzkonvertierung von Zeichensatz 1 nach Zeichensatz 2 durch. iconv --list liefert eine umfangreiche Liste aller unterstützten Zeichensätze.



### Beispiel

Das folgende Kommando erzeugt aus einer Latin-1-codierten Textdatei eine entsprechende UTF-8-Datei:

```
user$ iconv -f latin1 -t utf-8 latin1dat > utf8dat
```

### id

`id` gibt den Namen und die ID-Nummer des Benutzers, seiner primären Gruppe und der weiteren zugeordneten Gruppen an. Unter CentOS, Fedora und RHEL liefert das Kommando auch den SELinux-Kontext.

### Beispiel

Der Benutzer `kofler` hat die UID 1000, gehört der primären Gruppe `kofler` mit der GID 1000 an und ist Mitglied der Gruppe `wheel` mit der GID 10:

```
root# id
uid=1000(kofler) gid=1000(kofler) Gruppen=1000(kofler),10(wheel)
Kontext=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

```
if bedingung; then
    kommandos
[elif bedingung; then
    kommandos]
[else
    kommandos]
fi
```

`if` bildet Verzweigungen in `bash`-Scripts. Der Block nach `then` wird nur ausgeführt, wenn die Bedingung erfüllt ist. Andernfalls werden (beliebig viele optionale) `elif`-Bedingungen ausgewertet. Gegebenenfalls wird der ebenfalls optionale `else`-Block ausgeführt.

Als Bedingung können mehrere Kommandos angegeben werden. Nach dem letzten Kommando muss ein Strichpunkt folgen. Als Kriterium gilt der Rückgabewert des letzten Kommandos. Vergleiche und andere Tests können mit dem Kommando `test` durchgeführt werden. Statt `test` ist auch eine Kurzschreibweise in eckigen Klammern zulässig. Dabei muss aber nach `[` und vor `]` jeweils ein Leerzeichen angegeben werden.

```
ifconfig [-a]
ifconfig schnittstelle
ifconfig schnittstelle [optionen] [ip-adresse]
```

In der ersten Syntaxvariante liefert `ifconfig` Informationen über alle Netzwerkschnittstellen (ohne `-a` nur für aktive Schnittstellen, mit `-a` auch für noch nicht aktive Schnittstellen). In der zweiten Syntaxvariante zeigt `ifconfig` Informationen über die angegebene Netzwerkschnittstelle an. In der dritten Syntaxvariante richtet das Kommando eine neue Schnittstelle ein bzw. entfernt diese wieder. Die folgenden Optionen gelten nur für die dritte Syntaxvariante:

- ▶ `up/down`  
aktiviert bzw. deaktiviert die Schnittstelle.
- ▶ `mtu n`  
stellt den Parameter *maximum transfer unit* ein.
- ▶ `netmask n`  
gibt die Netzwerkmaske an. Das ist nur notwendig, wenn die Maske von der Standardmaske für die gewählte Adresse abweicht.

Beachten Sie bitte, dass das `ifconfig`-Kommando als veraltet gilt. Verwenden Sie stattdessen das Kommando `ip`, das in dieser Kommandoreferenz wesentlich ausführlicher beschrieben wird!

### Beispiel

Die beiden folgenden Kommandos aktivieren die Netzwerkschnittstelle `eth0` und weisen ihr die Adresse `192.168.0.2` zu. Für eine manuelle Einbindung des Rechners in ein lokales Netzwerk ist darüber hinaus auch eine Nameserver-Konfiguration in der Datei `/etc/resolv.conf` sowie die Einrichtung einer Default-Route mit dem Kommando `route` erforderlich.

```
root# ifconfig eth0 up
root# ifconfig eth0 192.168.0.2
```

```
iftop [optionen]
```

`iftop` beobachtet den Netzwerkverkehr einer bzw. aller Netzwerkschnittstellen und zeigt auf einer Seite, die alle drei Sekunden aktualisiert wird, zu welchen Hosts bzw. IP-Adressen die meisten Daten fließen. `iftop` läuft wie `top`, bis es mit `[q]` beendet wird.

- ▶ `-B`  
rechnet in Bytes/s statt in Bits/s.
- ▶ `-F ipadr/mask`  
berücksichtigt nur Verkehr von der bzw. zu der angegebenen Adresse.
- ▶ `-G ip6adr/mask`  
berücksichtigt nur Verkehr von der bzw. zu der angegebenen IPv6-Adresse.
- ▶ `-i name`  
berücksichtigt nur die angegebene Netzwerkschnittstelle.
- ▶ `-n`  
zeigt IP-Adressen statt Hostnamen.

`ifup` schnittstelle  
`ifdown` schnittstelle

`ifup` aktiviert die angegebene Schnittstelle, `ifdown` deaktiviert sie wieder. Die Kommandos werden vom Init-System zur Netzwerkinitialisierung aufgerufen und greifen auf die distributionsspezifischen Konfigurationsdateien zurück. Daher variiert die Implementierung der Kommandos je nach Distribution; auch die verfügbaren Optionen und deren Bedeutung hängen von der Distribution ab (siehe man `ifup/ifdown`).

### Beispiel

Die beiden folgenden Kommandos fahren die Netzwerkschnittstelle `eth0` zuerst herunter und dann wieder hoch – beispielsweise, um eine geänderte Konfiguration zu aktivieren:

```
root# ifdown eth0
root# ifup eth0
```

**info** [kommandoname]

`info` startet das gleichnamige Online-Hilfesystem. Zur Navigation im Hilfetext verwenden Sie die in [Tabelle 14](#) zusammengefassten Tastenkürzel. `info`-Texte können Sie alternativ auch mit dem Kommando `pinfo` aus dem gleichnamigen Paket, mit dem Editor Emacs oder in den Hilfesystemen von Gnome und KDE lesen. Alle Varianten bieten mehr Komfort als das Original.

► `-f datei`

lädt die angegebene Datei statt einer Datei aus `/usr/share/info`. Wenn der info-Text auf mehrere Dateien verteilt ist, muss die erste Datei angegeben werden (etwa `elisp-1.gz`).

### `init` [*n*]

`init` aktiviert den durch *n* angegebenen Runlevel. Das funktioniert sowohl bei Distributionen, die das traditionelle Init-V-System einsetzen, als auch bei Linux-Versionen, die auf Upstart setzen (z. B. CentOS 6, RHEL 6 und Ubuntu bis Version 14.10).

Bei Distributionen mit Systemd als Init-System verändern Sie den Runlevel bzw. genau genommen das »Target« mit dem Kommando `systemctl isolate`. Die Datei `/usr/sbin/init` ist in diesem Fall ein Link auf `systemd`.

### `initctl` [optionen] kommando

`initctl` steuert das Upstart-System, generiert Upstart-Ereignisse etc.

► `emit name`

erzeugt ein Ereignis mit dem angegebenen Namen.

► `list`

liefert Statusinformationen zu allen Upstart-Jobs.

► `start name`

startet den angegebenen Upstart-Job.

► `status name`

liefert Statusinformationen zum angegebenen Upstart-Job.

► `stop name`

stoppt den angegebenen Upstart-Job.

### `inotifywait` [optionen] [dateien/verzeichnisse]

`inotifywait` aus dem Paket `inotify-tools` überwacht die Veränderungen von Dateien bzw. deren Metadaten. In der einfachsten Form übergeben Sie an das Kommando einen oder mehrere Dateinamen. In diesem Fall wartet das Kommando, bis für

eine dieser Dateien ein inotify-Event auftritt, also z. B. eine Veränderung der Datei, ein Lesezugriff etc. Damit endet das Kommando. Das Kommando wird so häufig in Scripts eingesetzt, um automatisiert auf Änderungen von Dateien zu reagieren. Alternativ können Sie die Überwachung auch unbegrenzt durchführen und die aufgetretenen Ereignisse protokollieren.

► `-d` bzw. `-m`

arbeitet als Hintergrundprozess (`-d`, *daemon*) oder im Vordergrund (`-m`, *monitor*). `inotifywait` endet nun nicht beim ersten auftretenden Ereignis, sondern läuft, bis es explizit beendet wird, z. B. durch `kill` oder `[Strg]+[C]`.

► `-e event`

reagiert nur auf das angegebene Ereignis. Standardmäßig verarbeitet das Kommando alle inotify-Ereignisse. Die Option `-e` kann mehrfach angegeben werden, um mehrere Ereignisse auszuwählen. Zu den wichtigsten Ereignissen zählen `access`, `close`, `create`, `delete`, `modify`, `move` und `open`. Eine detaillierte Beschreibung aller Ereignisse gibt die `man`-Seite.

► `--fromfile datei.txt`

liest die Liste der zu überwachenden Dateien oder Verzeichnisse zeilenweise aus `datei.txt`.

► `-q`

verzichtet auf unnötige Ausgaben (*quiet*).

► `-r`

beobachtet rekursiv auch alle Unterverzeichnisse des angegebenen Startverzeichnisses. Dabei wird für jede einzelne Datei eine inotify-Überwachung eingerichtet. Bei Verzeichnissen mit vielen Dateien dauert das eine Weile und erfordert relativ hohe Ressourcen. Die Maximalanzahl der Überwachungen ist normalerweise mit 8192 festgelegt. Dieser Wert kann bei Bedarf in der Datei `/proc/sys/fs/inotify/max_user_watches` verändert werden.

► `-t n`

endet in jedem Fall nach *n* Sekunden, auch wenn kein Ereignis auftritt.

## Beispiel

Das folgende Shell-Script überwacht die Dateien `*.text` im aktuellen Verzeichnis. Bei jeder Veränderung in einer dieser Dateien überprüft es, ob es eine `*.text`-Datei gibt, die aktueller ist als die entsprechende `*.pdf`-Datei. In diesem Fall wird das betreffende PDF-Dokument mit `pandoc` neu erzeugt.

```
#!/bin/bash
while :
do
  for mdfile in *.text; do
    pdfoutfile=${mdfile%.text}.pdf
    if [ $mdfile -nt $pdfoutfile ]; then
      echo $mdfile
      pandoc -t beamer -H header.tex $mdfile -o $pdfoutfile
    fi
  done
  inotifywait -e modify -q *.text
done
```

**insmod** [optionen] moduldatei [parameter=wert ...]

insmod lädt das angegebene Kernelmodul. Dabei muss der vollständige Dateiname übergeben werden. Zusätzlich können Parameter (Optionen) an das Modul übergeben werden. Falls Sie hexadezimale Werte angeben möchten, müssen Sie 0x voranstellen, also etwa `option=0xff`. Die zur Auswahl stehenden Parameter des Moduls können Sie mit modinfo ermitteln.

► -f

versucht, das Modul selbst dann zu laden, wenn es nicht für die laufende Kernelversion kompiliert wurde. Ob das tatsächlich funktioniert, hängt davon ab, ob es zwischen der Kernel- und der Modulversion irgendwelche Inkompatibilitäten gibt. Die Option ist vor allem dann sinnvoll, wenn Hardware-Hersteller ein Modul nur als Binärversion (ohne Quellcode) zur Verfügung stellen. Die Option ist aber natürlich keine Garantie dafür, dass das Modul tatsächlich kompatibel zu Ihrer Kernelversion ist.

**insserv** kommando

insserv ist ein Kommando zum Einrichten bzw. Löschen von Init-V-Runlevel-Links. Es steht schon seit vielen Jahren unter SUSE zur Verfügung, seit Version 6 auch unter Debian. In aktuellen Distributionen mit Systemd ist insserv nur noch von untergeordneter Bedeutung: Es kann ausschließlich Init-Links manipulieren, die durch herkömmliche Init-V-Skripts gestartet werden. Es hat hingegen keinen Einfluss auf direkt durch Systemd kontrollierte Dienste.

insserv setzt voraus, dass im Kommentarblock am Beginn jedes Init-V-Skripts Informationen über abhängige Dienste enthalten sind. Jedes Mal, wenn Links für einen

zusätzlichen Init-V-Dienst eingerichtet bzw. Links entfernt werden, nummeriert `insserv` sämtliche Scripts neu und erzeugt die Dateien `.depend.*`, die für die parallele Ausführung der Scripts erforderlich sind. Aus diesem Grund muss bei SUSE jede Veränderung der Init-V-Runlevel-Links mit `insserv` erfolgen. Eine direkte Manipulation der Links ist nicht empfehlenswert.

`insserv` hat keine unmittelbare Auswirkung auf den angegebenen Dienst, sondern steuert nur, ob das Programm in Zukunft automatisch gestartet werden soll oder nicht. Um einen Systemdienst unmittelbar zu starten bzw. zu stoppen, führen Sie entweder das betreffende Init-V-Script direkt aus oder greifen auf das Kommando `service` zurück.

► *name*

richtet Links auf das angegebene Init-V-Script in den dafür vorgesehenen Standard-Runleveln ein. Auch diese Information geht aus dem Kommentarblock im Script hervor.

► `-r` bzw. `--remove name`

entfernt alle Links für das angegebene Init-V-Script.

## Beispiel

Die beiden folgenden Kommandos starten auf Init-V-basierten Distributionen den MySQL-Server unmittelbar und sorgen dafür, dass das Programm auch in Zukunft automatisch gestartet wird:

```
root# service mysqld start
root# insserv mysqld
```

Analog wird der MySQL-Server durch die beiden folgenden Kommandos heruntergefahren und auch in Zukunft nicht mehr gestartet:

```
root# service mysqld stop
root# insserv -r mysqld
```

**invoke.rc** *name* *kommando*

`invoke.rc` führt auf Debian-basierten Systemen ein Init-V-Script aus. `invoke.rc apache2 start` entspricht `/etc/init.d/apache2 start`. Das Kommando ist nur für Dienste geeignet, die nicht direkt durch `Systemd` kontrolliert werden.

### **ionice** [optionen] [kommando]

`ionice` führt das angegebene Kommando mit einer veränderten I/O-Priorität aus. `ionice` hat damit eine ähnliche Funktion wie `nice`, beeinflusst aber I/O-Operationen und nicht die CPU-Auslastung.

► `-c n`

gibt die gewünschte Scheduling-Klasse an. Zulässige Einstellungen sind:

- 0: keine Präferenzen
- 1: *realtime*, also maximale I/O-Geschwindigkeit
- 2: *best-effort*, gilt standardmäßig.
- 3: *idle*, also I/O-Operationen nur durchführen, wenn das System gerade nicht ausgelastet ist

► `-n n`

gibt die Prioritätsstufe innerhalb der gewählten Scheduling-Klasse an. Die zulässigen Werte reichen von 0 (maximale Priorität) bis 7 (minimale Priorität). Prioritätsstufen sind nur für die Scheduling-Klassen 1 und 2 vorgesehen und erlauben eine Differenzierung innerhalb der Klasse.

► `-p pid`

verändert die I/O-Priorität des durch die ID-Nummer angegebenen Prozesses.

### Beispiel

Das folgende Kommando startet ein Backup-Script mit minimaler I/O-Priorität:

```
root# ionice -c 3 backupscript
```

### **iotop**

`iotop` aus dem gleichnamigen Paket zeigt die I/O-Aktivität aller laufenden Prozesse an. Das hilft bei der Suche nach Prozessen, die die Festplatte oder andere Datenträger besonders stark beanspruchen.

► `-o`

zeigt nur Prozesse an, die tatsächlich I/O-aktiv sind (und nicht standardmäßig alle laufenden Prozesse).

► `-u` bzw. `--user=user`

zeigt nur die Prozesse des angegebenen Benutzers.



**ip** [optionen] objekt kommando

ip ist ein ungemein vielseitiges Kommando, um Informationen über Netzwerk-Devices, Tunnel, Routing-Regeln etc. zu ermitteln bzw. diese Einstellungen zu ändern. ip sollte anstelle von ifconfig und route verwendet werden, da diese beiden Kommandos als veraltet gelten.

► **-f fam** bzw. **-family fam**

bestimmt das gewünschte Netzwerkprotokoll (inet, inet6 oder link). Statt -f inet ist die Kurzschreibweise -4 zulässig, statt -f inet6 die Option -6, statt -f link die Option -0.

► **-o** bzw. **-oneline**

fasst zusammengehörende Ausgaben in einer Zeile zusammen. Das reduziert die Lesbarkeit, vereinfacht aber die Weiterverarbeitung durch grep oder wc.

► **-r** bzw. **-resolve**

löst IP-Adressen auf und zeigt stattdessen die Hostnamen an. Das erfordert einen Nameserver.

Als objekt muss eines der folgenden Schlüsselwörter angegeben werden: *addr*, *addr-label*, *link* (also eine Netzwerkschnittstelle), *maddr* (eine Multicast-Adresse), *mroute*, *monitor*, *neighbor* (ein ARP- oder NDISC-Cache-Eintrag), *route*, *rule* oder *tunnel*. Diese Schlüsselwörter dürfen abgekürzt werden. Für die meisten Objekte stehen die Kommandos *add*, *delete* und *list* = *show* zur Auswahl. Die weiteren Kommandos sind objektspezifisch. In der folgenden Referenz beschränke ich mich auf die wichtigsten Kommandos für die Objekte *addr*, *link* und *route*.

► **ip addr [show dev xxx]**

zeigt die IP-Adressen aller Schnittstellen. Die Ausgabe umfasst normalerweise mehrere Zeilen. Die mit *link/ether* beginnende Zeile gibt die MAC-Adresse der Schnittstelle an. Die mit *inet* beginnende Zeile enthält die IPv4-Adresse samt Maske in der Kurzschreibweise */n* sowie die Broadcast-Adresse. Die mit *inet6* beginnenden Zeilen geben die IPv6-Adressen an; das können mehrere sein.

Mit -4 oder -6 kann die Ausgabe auf IPv4 oder IPv6 eingeschränkt werden. *ip addr show dev xxx* liefert nur Informationen zur angegebenen Schnittstelle.

► **ip addr add n/m dev xxx**

fügt die IP-Adresse *n* mit der Maske *m* der Schnittstelle *xxx* hinzu. Eine zulässige IPv4-Adresse samt Maske wäre z. B. 10.0.45.34/24.

- ▶ `ip addr del n/m dev xxx`  
macht die Adresszuweisung zur Schnittstelle `xxx` rückgängig. Es müssen exakt dieselben Parameter wie bei `ip addr add` angegeben werden.
- ▶ `ip addr flush dev xxx`  
löscht *alle* Adresszuweisungen der Schnittstelle `xxx`.
- ▶ `ip link [show dev xxx]`  
liefert eine Liste aller Netzwerkschnittstellen, im Gegensatz zu `ip addr show` aber ohne die Angabe von IP-Adressen.
- ▶ `ip link set xxx up/down`  
aktiviert bzw. deaktiviert die Netzwerkschnittstelle.
- ▶ `ip neigh`  
liefert eine Liste aller anderen im lokalen Netzwerk bekannten IP-Adressen, also eine Aufzählung der »Nachbarn«.
- ▶ `ip route [list]`  
gibt die IPv4-Routing-Tabelle aus. Wenn Sie IPv6-Daten wünschen, müssen Sie die Option `-6` angeben. Die Gateway-Adresse geht aus der Zeile hervor, die mit `default` beginnt.
- ▶ `ip route add default via n`  
legt die IP-Adresse `n` als Default-Gateway fest.
- ▶ `ip route add n1/m via n2 dev xxx`  
definiert für den Adressbereich `n1/m` die Routing-Adresse `n2`. Die IP-Pakete werden über die Schnittstelle `xxx` geleitet.
- ▶ `ip route del ...`  
entfernt den angegebenen Routing-Eintrag. Die Parameter müssen exakt mit denen des Kommandos `ip route add` übereinstimmen.

Die Dokumentation des `ip`-Kommandos ist über mehrere man-Seiten verteilt. `man ip` gibt lediglich einen Überblick. `ip-address` liefert Details zu `ip addr`, `ip-route` zu `ip route` etc. Außerdem können Sie mit `ip objekt command help` eine Syntaxbeschreibung eines bestimmten Kommandos ermitteln, also beispielsweise mit `ip addr del` die Syntax zum Auflösen einer Adresszuordnung.

## Beispiel

Das folgende Kommando zeigt die aktuelle Routing-Tabelle an. Bei vielen modernen Distributionen kommen dabei anstelle von eth0 Schnittstellennamen wie enp0s3 zur Anwendung.

```
user$ ip route show
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.41 metric 1
default via 10.0.0.138 dev eth0 proto static
```

Um der Schnittstelle eth0 die Adresse 10.0.0.41 zuzuweisen und das Gateway 10.0.0.138 einzurichten, führen Sie die folgenden Kommandos aus:

```
root# ip route add 10.0.0.41/24 dev eth0
root# ip route add default via 10.0.0.138 dev eth0
```

Die folgenden Kommandos zeigen eine IPv6-Konfiguration:

```
root# ip -6 addr add 2a01:4f8:161:107::2/64 dev eth0
root# ip -6 route add default via fe80::1 dev eth0
```

Eine kompakte Liste aller Netzwerkschnittstellen liefert `ip -o link`:

```
root# ip -o link
1: lo:      <LOOPBACK,UP,LOWER_UP>          mtu 16436 ...
2: eth0:    <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
3: br0:     <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
4: virbr0:  <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 ...
5: vnet0:   <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 ...
```

```
ipcalc ipaddress [netmask]
```

ipcalc aus dem gleichnamigen Paket ermittelt aus einer gegebenen IPv4-Adresse und der Netzmaske alle weiteren Parameter, also die Netzwerkadresse, die Broadcast-Adresse etc.

```
user$ ipcalc 10.11.12.13/16
Address: 10.11.12.13      00001010.00001011. 00001100.00001101
Netmask: 255.255.0.0 = 16 11111111.11111111. 00000000.00000000
Wildcard: 0.0.255.255    00000000.00000000. 11111111.11111111

Network: 10.11.0.0/16     00001010.00001011. 00000000.00000000
HostMin: 10.11.0.1       00001010.00001011. 00000000.00000001
HostMax: 10.11.255.254   00001010.00001011. 11111111.11111110
Broadcast: 10.11.255.255 00001010.00001011. 11111111.11111111
Hosts/Net: 65534         Class A, Private Internet
```

```
iptables [optionen]
ip6tables [optionen]
```

iptables konfiguriert den Filter für Netzwerkpakete (kurz *Netfilter*) des Linux-Kernels. Die iptables-Optionen folgen einem einfachen Schema: Eine Option in Großbuchstaben gibt die durchzuführende Aktion an (beispielsweise `-P` zur Einstellung des Standardverhaltens). Weitere Optionen in Kleinbuchstaben steuern die Details dieser Aktion. Diese Syntaxzusammenfassung ist nach Aktionen gegliedert.

Beachten Sie, dass der Linux-Kernel vollkommen getrennte Filtertabellen für IPv4 und für IPv6 verwaltet. Dementsprechend gibt es auch zwei Konfigurationskommandos: iptables für IPv4 und ip6tables für IPv6. Die folgende Beschreibung gilt gleichermaßen für beide Kommandovarianten.

**iptables -P** chain policy [-t table]

iptables -P (*policy*) definiert das Standardverhalten für die angegebene Regelkette. Mögliche Verhalten sind:

ACCEPT:	Paket weiterleiten (Grundeinstellung)
DROP:	Paket löschen
RETURN:	Paket zurücksenden (selten)
QUEUE:	Paket an ein Programm außerhalb des Kernels weiterleiten (selten)

Standardmäßig gilt das Kommando für *Filter*-Regelketten oder für selbst definierte Regelketten. Falls eine *NAT*- oder *Mangle*-Regelkette verändert werden soll, muss der Tabellename mit der Option `-t` angegeben werden, z. B. `iptables -P POSTROUTING ACCEPT -t nat`.

Es ist nicht möglich, ein Standardverhalten für selbst definierte Regelketten zu definieren. Sie können das Standardverhalten aber bei Bedarf durch die letzte Regel definieren, z. B. durch `iptables -A mychain -j DROP`.

**iptables -A** chain [-t table] options

iptables -A (*add*) fügt der angegebenen Regelkette eine neue Regel hinzu. Generell gilt eine Regel für alle möglichen Fälle (d. h. für alle IP-Protokolle, für alle Ports, für alle Absender- und Zieladressen, für alle Interfaces etc.).

Durch Optionen kann die Gültigkeit eingeschränkt werden. Die meisten Optionen können mit einem Ausrufezeichen auch verneint eingesetzt werden. Mit `-p udp` gilt eine Regel also beispielsweise nur für UDP-Pakete. Mit `-p ! udp` gilt sie hingegen für alle Pakete außer für UDP-Pakete.

Nicht alle möglichen Kombinationen der Optionen sind zulässig. Beispielsweise dürfen die Optionen `-d` und `-s` nur für `tcp`-Pakete verwendet werden, also in Kombination mit `-p tcp`.

► `-d ipadresse`

gibt die Zieladresse an (*destination*). Adressbereiche können in der Form `192.168.0.0/24` oder `192.168.0.0/255.255.255.0` angegeben werden. In beiden Fällen sind alle IP-Nummern `192.168.0.*` gemeint.

► `--dport port[:port]`

gibt den Port oder Port-Bereich (z. B. `0:1023`) der Zieladresse an.

► `-i interface`

gibt das Interface an, aus dem das IP-Paket kommt (nur für *Input*-, *Forward*- und *Prerouting*-Regelketten). Beim Interface-Namen ist das Sonderzeichen `+` als Platzhalter für alle Interface-Nummern erlaubt, also `ppp+` für `ppp0`, `ppp1` etc.

► `-j ACCEPT/DROP/mychain/..`

gibt an, was mit dem Paket geschehen soll (*jump*). Hier wird meistens eines der vorgegebenen Verfahren (`ACCEPT`, `DROP` etc.) angegeben. Für Spezialanwendungen sieht `iptables REDIRECT` oder `MASQUERADE` vor.

Anstelle eines der vordefinierten Schlüsselwörter kann auch eine selbst definierte Regelkette angegeben werden. In diesem Fall werden alle Regeln dieser Kette angewandt. Falls keine Regel der selbst definierten Regelkette zutrifft, kommt die nächste Regel der ursprünglichen Regelkette zur Anwendung. In der prozeduralen Programmierung würde das einem Unterprogrammaufruf entsprechen.

► `-m module`

gibt an, dass ein Zusatzmodul verwendet werden soll. In der Folge dürfen spezielle Optionen verwendet werden, die durch dieses Zusatzmodul definiert sind. Ein besonders wichtiges Zusatzmodul ist `state`. Damit können Pakete nach ihrem Verbindungsstatus ausgewählt werden. Beispielsweise gilt eine Regel mit `-m state --state NEW` nur für IP-Pakete, die neue Verbindungen initiieren. Mit `--state` können folgende Statusschlüsselwörter angegeben werden:

NEW:	Das Paket initiiert eine neue Verbindung.
ESTABLISHED:	Das Paket gehört zu einer schon existierenden Verbindung.
RELATED:	Das Paket initiiert eine neue Verbindung, gehört aber zu einer schon existierenden Verbindung.
INVALID:	Das Paket gehört zu keiner vorhandenen Verbindung und initiiert auch keine neue Verbindung.

- ▶ `-o interface`  
gibt das Interface an, zu dem das IP-Paket unterwegs ist (nur für *Output*-, *Forward*- und *Postrouting*-Regelketten).
- ▶ `-p protocol`  
bestimmt das Protokoll (z. B. `tcp`, `udp` oder `icmp`).
- ▶ `-s ipadresse`  
gibt die Absenderadresse an (*source*).
- ▶ `--sport port[:port]`  
gibt den Port oder Port-Bereich für den Absender an.
- ▶ `--syn`  
gibt an, dass die Regel nur für solche TCP-Pakete gelten soll, bei denen das SYN-Bit gesetzt ist. Derartige Pakete werden verwendet, um eine Verbindung zu initiieren (etwa für alle TCP-Wrapper-Funktionen, für HTTP etc.).

iptables bietet die Möglichkeit, die Wirksamkeit einzelner Regeln durch `syslogd` zu protokollieren. Dazu geben Sie bei der Regel als Aktion `LOG` an. Damit eine Regel sowohl wirksam ist als auch protokolliert wird, muss sie zweimal angegeben werden: einmal mit `-j LOG` und ein zweites Mal mit `-j ACCEPT` bzw. `-j DROP`! Beachten Sie, dass durch Logging-Regeln sehr rasch riesige Protokolldateien entstehen können.

### **iptables -N** mychain

`iptables -N (new)` erzeugt eine neue Regelkette mit dem Namen *mychain*.

### **iptables -L** [chain] [-t table] [-v]

`iptables -L (list)` liefert ohne weitere Optionen eine Liste aller Regeln für die drei Regelketten der *Filter*-Tabelle sowie für alle selbst definierten Regelketten. Mit den weiteren Optionen können Sie die gewünschte Regelkette genau spezifizieren (z. B. `iptables -L mychain` oder `iptables -L POSTROUTING -t nat`). Die Zusatzoption `-v` bewirkt detailliertere Informationen. `-n` führt dazu, dass bei der Ausgabe IP- und Port-Nummern angezeigt werden (statt Netzwerk- bzw. Port-Namen).

### **iptables -D** chain [-t table] options

`iptables -D (delete)` löscht die Regel aus der Regelkette. Es müssen exakt dieselben Optionen wie bei `iptables -A` angegeben werden.

**iptables -F** chain [-t table]

iptables -F (*flush*) löscht alle Regeln aus der angegebenen Regelkette.

**iptables -X** [mychain]

iptables -X löscht die angegebene eigene Regelkette. Wenn keine Regelkette angegeben wird, werden alle selbst definierten Regelketten gelöscht.

### Beispiel

Die folgenden iptable-Kommandos definieren eine Mini-Firewall für IPv4. Dabei werden eintreffende Pakete nur dann akzeptiert, wenn sie entweder einer bereits existierenden Verbindung zuzuordnen sind oder *nicht* von der Schnittstelle eth0 stammen, über die der Rechner mit dem Internet verbunden ist. Das Beispiel geht davon aus, dass sich die iptables-Tabellen anfänglich im Defaultzustand befinden, also alle Pakete akzeptieren, und dass es nur eine Schnittstelle zum Internet gibt.

```
root# iptables -N wall
root# iptables -A wall -m state --state ESTABLISHED,RELATED -j ACCEPT
root# iptables -A wall -m state --state NEW ! -i eth0 -j ACCEPT
root# iptables -A wall -j DROP
root# iptables -A INPUT -j wall
root# iptables -A FORWARD -j wall
```

```
ip[6]tables-save [optionen]
ip[6]tables-restore [optionen]
ip[6]tables-xml [optionen]
```

Die Kommandos iptables-xxx bzw. ip6tables-xxx helfen dabei, die Regeln einer Paketfilter-Firewall in einer Datei zu speichern bzw. daraus zu lesen. Jedes im Folgenden kurz beschriebene Kommando gibt es auch in einer IPv6-Variante.

- ▶ iptables-save gibt alle Regeln aller Firewall-Filter aus. Dabei wird eine gut lesbare Syntax verwendet. Die Ausgabe kann mit > in eine Datei umgeleitet werden. Die Option -t name bewirkt, dass nur die Regeln des angegebenen Filters gespeichert werden.
- ▶ iptables-restore liest Regeln aus der Standardeingabe. Die Regeln ersetzen normalerweise bereits vorhandene Filter-Regelketten. Wenn Sie das nicht möchten, geben Sie die Option -n an. Mit -T name werden nur die Regeln des angegebenen Filters berücksichtigt.
- ▶ iptables-xml funktioniert ähnlich wie iptables-save, erzeugt aber ein XML-Dokument.

### isohybrid datei.iso

Das Kommando `isohybrid` aus dem Paket `syslinux` manipuliert ein bootfähiges ISO-Image so, dass das BIOS das Image auch von einem USB-Stick oder einer USB-Festplatte booten kann.

#### Beispiel

Die beiden folgenden Kommandos wandeln das ISO-Image zuerst in den Hybrid-Modus um, sodass ein Systemstart sowohl von einer CD/DVD als auch von einem USB-Datenträger möglich ist. Anschließend überträgt `dd` die Image-Datei auf einen USB-Stick, der über den Device-Namen `/dev/sdb` angesprochen wird. Stellen Sie unbedingt sicher, dass Sie beim `dd`-Kommando mit `of` nicht irrtümlich Ihre lokale Festplatte angeben und diese so löschen!

```
user$ isohybrid mylinux.iso
root# dd if=mylinux.iso of=/dev/sdb bs=1M
```

### iw objekt kommando

Mit dem Kommando `iw` steuern Sie WLAN-Adapter, die die `nl80211`-Schnittstelle unterstützen. Das ist bei den meisten aktuellen WLAN-Adaptoren der Fall, deren Treiber auf dem `mac80211`-Framework basieren.

Es gibt mehrere Möglichkeiten, um das Objekt anzugeben, das Sie steuern wollen. Dabei darf das Kürzel `dev` bzw. `phy` weggelassen werden, wenn der Schnittstellen- oder Gerätenamen eindeutig ist.

- ▶ `dev name`  
gibt den Schnittstellennamen an (z. B. `dev wlan0`).
- ▶ `phy name` bzw. `phy #n`  
gibt den Namen bzw. die Indexnummer des Geräts an. Bei Notebooks mit einem WLAN-Adapter lautet der Gerätenamen immer `phy0`.
- ▶ `reg`  
steuert den *regulatory agent*, also ein Regelwerk für nationale Funkstandards.

Die zur Auswahl stehenden Kommandos hängen vom Objekttyp ab. Im Folgenden stelle ich nur einige ausgewählte Kommandos vor:

- ▶ `dev name connect ssid`  
stellt eine Verbindung zum angegebenen WLAN-Netzwerk her. Das gelingt nur bei Netzwerken ohne Verschlüsselung. Wenn das Funknetzwerk durch WEP abge-



sichert ist, geben Sie den Schlüssel durch den optionalen Parameter `keys` an (z.B. `keys 0:0011223344`). Der Schlüssel wird wahlweise in Form von 5 oder 13 ASCII-Zeichen bzw. durch 10 oder 26 hexadezimale Ziffern angegeben. Wenn das Netzwerk durch WPA abgesichert ist, müssen Sie den Schlüssel vorweg an das Kommando `wpa_supplicant` übergeben.

- ▶ `dev name del`  
entfernt (löscht) die Schnittstelle. Wenn die Schnittstelle später wieder verwendet werden soll, muss sie mit `interface add` neu eingerichtet werden.
- ▶ `dev name disconnect`  
beendet die Verbindung.
- ▶ `dev name info`  
gibt allgemeine Informationen über die Schnittstelle an.
- ▶ `dev name link`  
liefert Informationen zur aktiven Netzwerkverbindung bzw. *not connected*.
- ▶ `dev name scan`  
liefert detaillierte Informationen zu allen in Reichweite befindlichen Funknetzen.
- ▶ `phy phy0 interface add wlan0 type managed`  
richtet die Schnittstelle `wlan0` für das Gerät `phy0` ein. Andere Schnittstellentypen sind `monitor`, `wds`, `mesh` bzw. `mp` sowie `ibss` bzw. `adhoc`. Die neue Schnittstelle muss anschließend mit `ifconfig wlan0 up` aktiviert werden.

Zur Fehlersuche ist es häufig zweckmäßig, in einem zweiten Fenster oder einer Konsole das Kommando `iw event` auszuführen. Es liefert bis zum Ende durch `Strg+C` alle Status- und Fehlermeldungen.

## Beispiel

Die folgenden Kommandos stellen manuell eine Netzwerkverbindung zu einem Funknetz her, das nicht durch ein Passwort geschützt ist:

```
root# iw phy phy0 interface add wlan0 type managed
root# ifconfig wlan0 up
root# iw dev wlan0 connect hotel-wlan
root# dhclient wlan0
```

Bei vielen aktuellen Distributionen heißen die WLAN-Schnittstellen nicht mehr `wlan0`, `wlan1` etc., sondern `wlpnsm`.

```
iwconfig [schnittstelle]
iwconfig schnittstelle [optionen]
```

Das Kommando `iwconfig` zählt zu den veralteten Linux-Wireless-Tools. Das Kommando wird dennoch bei den meisten Linux-Distributionen standardmäßig installiert. Nach Möglichkeit (d. h., wenn es für Ihren WLAN-Adapter einen modernen Treiber gibt, der die nl80211-Schnittstelle unterstützt) sollten Sie aber das neuere Kommando `iw` vorziehen.

In der ersten Syntaxvariante liefert `iwconfig` Informationen über alle WLAN-Schnittstellen bzw. über die angegebene Schnittstelle. In der zweiten Syntaxvariante stellt das Kommando die Parameter der WLAN-Schnittstelle ein (z. B. den Netzwerknamen, den WEP-Schlüssel etc.). `iwconfig` ist Teil der Wireless-Tools. Die eigentliche Aktivierung der Schnittstelle erfolgt anschließend wie bei LAN-Schnittstellen durch `ip` oder `ifconfig`.

- ▶ `channel n`  
wählt den Frequenzkanal aus. Mit `channel auto` sucht der WLAN-Controller selbst einen geeigneten Kanal. Eine Kanalliste liefert `iwlist channel`.
- ▶ `essid name`  
gibt den Namen des WLAN-Netzes an. Oft funktioniert auch die Einstellung `any`.
- ▶ `key schlüssel`  
stellt den aktuellen WEP-Schlüssel ein. Der Schlüssel wird normalerweise als hexadezimale Zahl ohne vorangestelltes `0x` angegeben. Mit `key [n]` wählen Sie den gerade aktuellen Schlüssel, wobei *n* zwischen 1 und 4 liegt. `ifconfig key` ist ungeeignet, um WPA-Schlüssel einzustellen! Der WPA-Schlüsselaustausch erfolgt durch das Hinterprogramm `wpa_supplicant`, das separat konfiguriert werden muss.
- ▶ `mode modus`  
bestimmt den Netzwerkmodus. Zur Auswahl stehen je nach Hardware `Managed`, `Ad-Hoc`, `Master`, `Repeater`, `Secondary`, `Monitor` oder `Auto`. Wenn Sie mit Ihrem WLAN-Controller auf einen WLAN-Router oder -Access-Point zugreifen möchten, lautet die richtige Einstellung `Managed`.

```
iwlist [schnittstelle] modus
```

`iwlist` liefert für alle bzw. für die angegebene WLAN-Schnittstelle die möglichen Frequenzkanäle, die zulässigen Verschlüsselungsverfahren etc. Die wichtigsten Schlüsselwörter für `modus` sind:

channel	Frequenzkanäle
frequency	Frequenzen
key	zulässige Verschlüsselungsverfahren und eingestellte Schlüssel
rate	unterstützte Bruttoübertragungsraten des WLAN-Controllers
scan	Liste der erreichbaren Netze mit ESSID, Qualität, Frequenz etc.

## j verzeichnis

Das Kommando `j` aus dem Paket `autojump` hilft dabei, besonders effizient in ein anderes Verzeichnis zu wechseln. `j` ist gewissermaßen eine mitlernende Variante zum `cd`-Kommando. Wenn Sie beispielsweise einmal `j /etc/X11/xorg.conf.d` ausgeführt haben, reicht beim zweiten Mal `j xorg.conf.d`, also die Angabe des letzten Teils des Verzeichnispfads (vorausgesetzt, dieser ist eindeutig). Mit der Vervollständigung durch `[Tab]` können Sie die Eingabe weiter verkürzen, z. B. in der Form `j xorg [Tab]`. Wenn mehrere Verzeichnisse passen, drücken Sie einfach mehrfach `[Tab]`. Eine Statistik aller zuletzt besuchten Verzeichnisse liefert bei Bedarf das Kommando `jumpstats`.

## journalctl [optionen] [suchausdruck]

Viele Distributionen, die Systemd als Init-System verwenden, haben auch das traditionelle Syslog-System durch eine Systemd-Implementierung ersetzt, das sogenannte Journal. Die Logging-Dateien werden dabei in einem speziellen binären Format gespeichert und können nur noch mit dem hier beschriebenen `journalctl`-Kommando ausgelesen werden.

Ein oder mehrere Suchausdrücke können als Pfad zu einem Programm oder in der Syntax `field=value` formuliert werden. `journalctl` zeigt dann nur Logging-Einträge, auf die alle Suchausdrücke zutreffen. Suchausdrücke mit logischem Oder können Sie mit `+` formulieren, also `field1=value1 + field2=value2`.

Für `field` sind unter anderem die Schlüsselwörter `MESSAGE`, `PRIORITY`, `ERRNO` sowie `_PID`, `_UID`, `_GID` oder `_SELINUX_CONTEXT` zulässig. Eine Referenz weiterer Suchparameter sowie eine genaue Beschreibung ihrer Bedeutungen können Sie mit `man systemd.journal-fields` nachlesen.

Bei einem Aufruf ohne weitere Parameter liefert `journalctl` eine zumeist fast endlose Liste aller protokollierten Meldungen. Mit Optionen können Sie die Ausgabe filtern:

### ► -b

zeigt nur die Nachrichten seit dem letzten Neustart des Rechners.

- ▶ `--disk-usage`  
zeigt an, wie viel Platz die Dateien des Journals in `/var/log` beanspruchen.
- ▶ `-e`  
springt sofort an das Ende der anzuzeigenden Nachrichten.
- ▶ `-f`  
startet `journalctl` im Dauerbetrieb, wobei ständig die gerade eintreffenden Nachrichten angezeigt werden. `[Strg]+[C]` beendet das Kommando.
- ▶ `-k`  
zeigt nur Kernelnachrichten.
- ▶ `-n n`  
zeigt nur die letzten *n* Zeilen.
- ▶ `--no-pager`  
schreibt die Ausgaben direkt auf die Standardausgabe, anstatt einen Pager (normalerweise das Kommando `less`) zur Anzeige zu verwenden.
- ▶ `-p n`  
zeigt nur Nachrichten in einer bestimmten Prioritätsstufe. Der Zahlenbereich reicht von 0 bis 7 für `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info` und `debug`. `-p 2` bewirkt, dass nur Nachrichten der Stufen 0, 1 und 2 berücksichtigt werden.
- ▶ `-r`  
zeigt die neuesten Nachrichten zuerst.
- ▶ `--since 2015-12-31 19:30:00`  
zeigt nur Nachrichten, die nach dem angegebenen Zeitpunkt protokolliert wurden.
- ▶ `-t name`  
zeigt nur Nachrichten für das angegebene Syslog-Stichwort (Tag, wie bei `logger -t`).
- ▶ `-u name`  
zeigt nur Nachrichten für den angegebenen Systemd-Dienst (Unit, z.B. `avahi-daemon`).
- ▶ `--until 2015-12-31 19:30:00`  
zeigt nur Nachrichten, die bis zum angegebenen Zeitpunkt protokolliert wurden.

## Beispiele

Das folgende Kommando zeigt alle Meldungen des Open-SSH-Servers in umgekehrter Reihenfolge, d. h., die neuesten Nachrichten zuerst:

```
user$ journalctl -u sshd -r
```

Wenn Sie die Logging-Nachrichten live verfolgen möchten, rufen Sie `journalctl` wie folgt auf:

```
user$ journalctl -u sshd -f
```

Das dritte Beispiel zeigt alle Meldungen an, in denen die IP-Adresse 10.0.0.2 vorkommt:

```
user$ journalctl | grep 10.0.0.2
```

```
kill [-s signal] prozessnr
```

Das bash-Kommando `kill` versendet Signale an einen laufenden Prozess. Wenn `kill` ohne die `-s`-Option verwendet wird, wird standardmäßig das SIGTERM-Signal (15) gesendet, um den Prozess zu beenden (zu *killen*, daher auch der Name des Kommandos). Bei besonders hartnäckigen Fällen hilft `-9` bzw. `-s SIGKILL` oder `-KILL`. Der Prozess hat dann allerdings keine Chance, noch irgendwelche Aufräumarbeiten zu erledigen.

`kill` kann aber auch zum Versenden harmloserer Signale verwendet werden. Recht häufig wird `-1` bzw. `-s SIGHUP` bzw. `-HUP` verwendet, um einen Dämon dazu aufzufordern, seine Konfigurationsdateien neu einzulesen. Auf diese Weise können Sie bei manchen Programmen eine neue Konfiguration aktivieren, ohne den Dämon vollständig stoppen und neu starten zu müssen.

Die erforderliche Prozessnummer (PID) wird am einfachsten mit dem Kommando `ps` ermittelt. Unter X gibt es mit `xkill` eine bequeme Variante zu `kill`: Das Programm, das beendet werden soll, kann damit einfach per Maus »abgeschossen« werden.

```
killall [-signal] prozessname
```

`killall` funktioniert beinahe wie das `kill`-Kommando. Der Unterschied besteht darin, dass nicht die Prozessnummer (PID), sondern der Name des Prozesses angegeben wird. Wenn es mehrere Prozesse dieses Namens gibt, erhalten alle das angegebene Signal (standardmäßig wieder SIGTERM). Das gewünschte Signal wird entweder als Nummer `-n` oder mit einem Namen wie `-HUP` angegeben. Eine Liste aller Signalnamen erhalten Sie mit `killall -l`.

### Beispiel

Das folgende Beispiel beendet alle laufenden Firefox-Instanzen des aktuellen Benutzers. Wird das `killall`-Kommando von `root` ausgeführt, beendet es alle laufenden Firefox-Prozesse *aller* Benutzer.

```
user killall firefox
```

### kpartx [diskdevice]

Das Low-Level-Kommando `kpartx` aus dem gleichnamigen Paket ermittelt alle Partitionen des angegebenen Datenträgers und erzeugt die dazugehörigen Device-Dateien. Normalerweise werden die Device-Dateien durch das `udev`-System automatisch erzeugt, sobald ein neuer Datenträger erkannt wird – z. B. beim Anschließen einer USB-Festplatte. `kpartx` ist primär zur Bearbeitung von virtuellen Datenträgern bzw. von Image-Dateien virtueller Maschinen gedacht.

- ▶ `-a`  
erzeugt neue Device-Dateien für den angegebenen Datenträger (*add*).
- ▶ `-d`  
entfernt die Device-Dateien für den Datenträger (*delete*).
- ▶ `-l`  
liest die Partitionen des Datenträgers, erzeugt aber keine Device-Dateien.
- ▶ `-u`  
aktualisiert die Device-Dateien für einen veränderten Datenträger (*update*).
- ▶ `-v`  
gibt Informationen über die durchgeführten Aktionen aus.

### Beispiel

Das folgende Kommando verbindet alle in der RAW-Image-Datei enthaltenen Partitionen mit Loop-Devices:

```
root# kpartx -av image.raw
add map loop0p1 (252:12): 0 1024000 linear /dev/loop0 2048
add map loop0p2 (252:13): 0 19945472 linear /dev/loop0 1026048
```

Die ganze virtuelle Festplatte kann anschließend über das Device `/dev/mapper/loop0` angesprochen werden.

```
kvm [optionen] [imagedatei]
qemu-kvm [optionen] [imagedatei]
```

kvm (Debian, Ubuntu) bzw. qemu-kvm (Fedora, Red Hat) führt eine virtuelle Maschine aus. Unter RHEL befindet sich das Kommando `qemu-kvm` im Verzeichnis `/usr/libexec/`.

Wenn Sie den Datenträger der Image-Datei am Ende des KVM-Kommandos angeben, emuliert KVM eine IDE-Festplatte. Wünschen Sie andere Einstellungen oder mehrere Datenträger, verwenden Sie dazu die Optionen `-drive` oder `-hda`, `-hdb` etc. In diesem Fall kann die direkte Angabe der Image-Datei im KVM-Kommando entfallen.

► `-boot order=xxx,once=xxx,menu=on/off`

gibt an, in welcher Reihenfolge die Datenträger für den Bootprozess berücksichtigt werden sollen. Dabei ist `xxx` eine Buchstabenfolge, die die Reihenfolge der Datenträger ausdrückt (z. B. `adc`: zuerst das Diskettenlaufwerk, dann das CD/DVD-Laufwerk, danach die erste Festplatte). Die Buchstaben `a` bis `d` entsprechen den DOS/Windows-Laufwerksbuchstaben.

`once=xxx` gibt die Bootreihenfolge nur für den *ersten* Bootvorgang an. Wenn die virtuelle Maschine also beispielsweise beim ersten Versuch vom CD/DVD-Laufwerk booten soll, bei weiteren Neustarts aber von der Festplatte, geben Sie `-boot order=c,once=d` oder schlicht `-boot once=d` an.

`menu=on` zeigt zum Beginn des Bootmenüs die Meldung *Press F12 for boot menu* an. Mit `[F12]` kann dann der Bootdatenträger interaktiv ausgewählt werden.

► `-cdrom iso-datei`

verwendet die angegebene ISO-Datei als Datenquelle für das virtuelle CD/DVD-Laufwerk. Die Option entspricht `-drive file=iso-datei,index=2,media=cdrom`.

► `-cpu host`

gibt alle Eigenschaften der Host-CPU an den Gast weiter. Standardmäßig ist das nicht der Fall: Es wird nur ein Subset weitergegeben, um die Kompatibilität virtueller Maschinen zwischen unterschiedlichen CPUs zu maximieren. Wenn Sie auf einem 64-Bit-Host arbeiten, dem Gast aber nur eine 32-Bit-CPU zur Verfügung stellen möchten, verwenden Sie `-cpu kvm32`.

► `-device gerät`

fügt der virtuellen Maschine ein zusätzliches Gerät hinzu. Eine Liste aller unterstützten Geräte liefert `kvm -device ?`. Beim Gerätenamen wird zwischen Groß- und Kleinschreibung unterschieden! Die für ein bestimmtes Gerät verfügbaren Optionen ermitteln Sie mit `kvm -device gerät,?, also z.B. kvm -device isa-serial,?`.

Beachten Sie, dass Sie die meisten Komponenten einer virtuellen Maschine auf zwei Arten definieren können: Mit der hier beschriebenen, sehr universellen Option `-device` oder mit gerätespezifischen Optionen (z. B. `-drive`, `-soundhw`, `-usb-device` oder `-vga`).

► `-drive details`

definiert die Eigenschaften einer virtuellen Festplatte. Die Detailparameter werden nur durch Kommata voneinander getrennt (ohne Leerzeichen!). Die Option kann mehrfach verwendet werden, wenn die virtuelle Maschine mit mehreren Datenträgern ausgestattet werden soll.

`if=ide/scsi/virtio` gibt an, über welche Schnittstelle die virtuelle Maschine den Datenträger sehen soll (standardmäßig `ide`). Bei Linux-Gästen ist `virtio` effizienter.

`media=disk/cdrom` gibt an, ob es sich um eine Festplatte (gilt standardmäßig) oder um ein CD/DVD-Laufwerk handeln soll.

`index=n` bestimmt die Nummerierung der Datenträger einer Schnittstelle. Der Parameter ist nur erforderlich, wenn die Datenträger nicht der Reihe nach angegeben werden.

`file=fname` gibt den Dateinamen der Image- oder ISO-Datei bzw. den Device-Namen eines Logical Volumes an.

`boot=on/off` gibt an, ob der Datenträger beim Booten berücksichtigt werden soll. Bei IDE- und SCSI-Laufwerken gilt automatisch `boot=on`. Damit KVM auch von einem `virtio`-Laufwerk booten kann, muss `boot=on` explizit angegeben werden.

`cache=writethrough/writeback/none` gibt an, ob und wie Schreibzugriffe zwischengespeichert werden. Standardmäßig gilt `writethrough`: Im Gastsystem erscheint ein Schreibzugriff erst dann als abgeschlossen, wenn das Hostsystem den Speichervorgang quittiert hat.

► `-enable-kvm`

aktiviert KVM. Bei vielen Distributionen ist diese Option automatisch aktiv, aber nicht bei allen! Ohne diese Option wird die virtuelle Maschine ohne CPU-Unterstützung emuliert, was ineffizient und langsam ist.

► `-hda/-hdb/-hdc/-hdd details`

gibt eine virtuelle IDE-Festplatte an.

`-hda fname` entspricht `-drive file=fname,index=0,media=disk`,

`-hdb fname` entspricht `-drive file=fname,index=1,media=disk` etc.



► `-k sprachkürzel`

verwendet das angegebene Tastaturlayout. Zulässige Kürzel sind unter anderem `de` (Deutsch) und `en-us` (US-Englisch). Die Option ist nur erforderlich, wenn die virtuelle Maschine durch einen externen VNC-Client bedient wird. Die VNC-Clients des Virtual Machine Managers bzw. des Kommandos `virt-viewer` erkennen die Tastatureinstellung selbstständig.

► `-localtime`

initialisiert die virtuelle CMOS-Uhr des Gastsystems mit der lokalen Zeit (statt standardmäßig mit der UTC-Zeit).

► `-m n`

stellt die Speichergröße der virtuellen Maschine ein (in MByte). Die Defaulteinstellung variiert je nach Distribution.

► `-monitor device`

leitet die Ein- und Ausgabe des QEMU-Monitors in das angegebene Device um. Wenn Sie den Monitor über die aktuelle Konsole bedienen möchten, geben Sie als Device `stdio` an. Mit `pty` legt `kvm` beim Start ein neues Pseudo-TTY-Device an und verwendet es für die Kommunikation.

► `-net nic,details`

konfiguriert einen virtuellen Netzwerkadapter. Wenn diese Option nicht angegeben wird, emuliert KVM standardmäßig eine RTL-8139-kompatible Netzwerkkarte.

`model=ne2k_pci/i82551/i82557b/i82559er/rtl8139/e1000/pcnet/virtio` legt fest, welchen Netzwerkadapter KVM emulieren soll. Für Linux-Gäste erzielen Sie mit `model=virtio` die besten Resultate. `macaddr=52:54:00:nn:nn:nn` gibt die gewünschte MAC-Adresse an.

► `-net user,details`

verwendet Usermode-Networking (gilt standardmäßig): Das Gastsystem kann zwar dank NAT und Masquerading die Internetverbindung des Hostsystems nutzen, es ist aber keine direkte Netzwerkverbindung zwischen Gast und Host möglich.

► `-nographic`

startet die virtuelle Maschine ohne Grafiksystem. Die Kommunikation mit der virtuellen Maschine erfolgt direkt in der aktiven Konsole über die serielle Schnittstelle des Gastsystems. Die Option `-nographic` ermöglicht also die Nutzung einer virtuellen Maschine im Textmodus, setzt aber voraus, dass im Gastsystem ein `getty`-Prozess mit der seriellen Schnittstelle kommuniziert.

► `-rtc base=utc/localtime`

gibt an, welche Startzeit die Uhr der virtuellen Maschine haben soll. `utc` ist die korrekte Einstellung für Linux-Gäste, während `localtime` für Windows-Gäste geeignet ist. Standardmäßig ist die Uhr immer synchron mit jener des Hostrechners. Wenn Sie das nicht wünschen, können Sie den zusätzlichen Parameter `clock=vm` angeben.

► `-smp n` bzw. `-smp cores=c,threads=t,sockets=s`

gibt in der Kurzform an, wie viele CPU(-Cores) der virtuellen Maschine zugewiesen werden sollen (standardmäßig nur ein Core). Bei Hostsystemen mit mehreren CPUs gibt `c` an, wie viele Cores pro CPU genutzt werden sollen. `t` gibt die gewünschte Anzahl der Threads pro Core an; sinnvoll ist hier zumeist der Wert 2 bei Intel-CPU, die Hyperthreading unterstützen. `s` legt schließlich fest, wie viele CPUs (Sockets) verwendet werden sollen.  $c * t * s$  ergibt die Anzahl der CPUs, die die virtuelle Maschine sieht.

► `-soundhw ac97/es1370/hda/sb16/all`

fügt der virtuellen Maschine eines der angegebenen Audio-Geräte hinzu (oder alle, wenn Sie `all` verwenden). `ac97` steht für Intel 82801AA AC97, `es1370` für Ensoniq AudioPCI ES1370, `hda` für Intel High Definition Audio und `sb16` für Creative Sound Blaster 16.

► `-spice port=n[,optionen]`

aktiviert das Grafiksystem Spice. Dazu muss zumindest der gewünschte Port angegeben werden. Mit `password=xxxx` kann die Verbindung zudem durch ein Passwort abgesichert werden. Wenn kein Passwort verwendet werden soll, muss explizit die Option `disable-ticketing` angegeben werden.

► `-usbdevice mouse/tablet/disk/host...`

fügt der virtuellen Maschine ein USB-Gerät hinzu. Am häufigsten werden Sie die Option `-usbdevice tablet` benötigen. Sie ersetzt die standardmäßig emulierte PS/2-Maus durch ein virtuelles USB-Zeigegerät, das absolute Koordinaten versteht und so die Synchronisation der Mausposition des Gasts mit dem VNC- oder Spice-Client ermöglicht.

`-usbdevice disk` ermöglicht es, eine Image-Datei des Hosts so an den Gast weiterzugeben, dass dieser einen USB-Datenträger sieht.

`-usbdevice host:bus.addr` bzw. `-usbdevice host:vendorid:productid` leitet ein USB-Gerät des Hosts an den Gast weiter. Das USB-Gerät darf vom Host nicht genutzt werden. Die Bus- und Device-Nummer bzw. die Vendor- und Produkt-IDs ermitteln Sie auf dem Hostrechner am einfachsten mit `lsusb`.

► `-vga cirrus/qxl/std/vmware`

gibt den gewünschten Typ der virtuellen Grafikkarte an. Standardmäßig emuliert KVM eine Cirrus-kompatible Grafikkarte mit einer Auflösung von bis zu  $1024 \times 768$  Pixeln. Dieses Grafiksystem wird von nahezu allen Gastsystemen korrekt erkannt und in einer akzeptablen Geschwindigkeit ausgeführt. Die Grafikkarte qxl kann nur in Kombination mit `-spice` eingesetzt werden.

► `-vnc n.n.n.n:n[,optionen]`

führt einen VNC-Server aus, über den Clients den Inhalt der virtuellen Grafikkarte darstellen können. Mit `n.n.n.n` geben Sie an, von welcher IP-Adresse aus der Verbindungsaufbau zum VNC-Server erfolgen darf (z. B. 127.0.0.1 für Verbindungen von localhost). `:n` gibt die Display-Nummer an. Der Port für den VNC-Server ergibt sich aus `n+5900`. Wenn Sie nur die Display-Nummer ohne IP-Adresse angeben (also z. B. `:0`), kann der Verbindungsaufbau durch jeden beliebigen Rechner erfolgen.

## Beispiel

Im folgenden Beispiel wird zuerst eine 10 GByte große Image-Datei erzeugt, die dann als virtuelle Festplatte zur Installation eines Ubuntu-Servers von einer ISO-Datei genutzt wird:

```
root# qemu-img create -f qcow2 disk.img 10G
root# kvm -enable-kvm -m 1024 -smp 2 -boot once=d -cdrom ubuntu-server.iso \
    -drive file=disk.img,if=virtio,format=qcow2 \
    -net user -net nic,macaddr=52:54:00:12:e4:4e,model=virtio \
    -vga cirrus -vnc 127.0.0.1:0 -k de -usb -usbdevice tablet
```

Um die virtuelle Maschine zu bedienen, müssen Sie nun noch einen VNC-Client starten, z. B. das Programm `vncviewer`:

```
user$ vncviewer localhost:0
```

## **l2ping** [optionen] bluetoothmac

`l2ping` aus dem `bluez`-Paket sendet L2CAP-Echo-Anfragen an das durch seine MAC-Adresse angegebene Bluetooth-Gerät. Auf diese Weise kann getestet werden, ob prinzipiell eine Verbindung zwischen dem lokalen Bluetooth-Adapter und dem externen Gerät möglich ist. Die MAC-Adresse externer Bluetooth-Geräte ermitteln Sie mit `hcitool scan`. `l2ping` muss mit `root`-Rechten ausgeführt werden.

```
lame [optionen] in out.mp3
```

Das Acronym LAME steht für *LAME Ain't an MP3 Encoder*. Tatsächlich dient das Kommando `lame` aber sehr wohl dazu, WAV-Dateien in komprimierte Audio-Dateien umzuwandeln, die zum MP3-Format kompatibel sind. Aufgrund von Patentschwierigkeiten steht `lame` bei vielen Distributionen nicht standardmäßig zur Verfügung und muss extra installiert werden, oft aus nichtoffiziellen Paketquellen. Alternativen zu `lame` sind `toolame` oder `twolame`, die beide Dateien im Format MPEG-1 Layer 2 (also MP2) erzeugen. Auch derartige Dateien werden von den meisten MP3-Playern problemlos abgespielt.

► `-r`

die Quelldatei liegt im RAW-Format vor (nicht als WAV-Datei).

► `-s 8/11.025/12/16/22.05/24/32/44.1/48`

gibt die Sampling-Frequenz innerhalb der RAW-Datei an. Die Option ist bei WAV-Dateien nicht erforderlich.

### Beispiel

Das folgende Kommando erzeugt die MP3-kompatible Audio-Datei `title.mp3`:

```
user$ lame title.wav title.mp3
```

```
last [optionen]
```

`last` liefert eine Liste der Benutzer, die zuletzt auf diesem Rechner eingeloggt waren. Das Kommando wertet die Datei `/var/log/wtmp` aus.

► `-i`

zeigt bei SSH-Logins die IP-Adresse anstelle des Hostnamens des entfernten Rechners an.

► `-n` oder `-n n`

steuert, wie viele Einträge angezeigt werden sollen, z. B. 100 mit `-100`.

► `-t YYYYMMDDhhmmss`

gibt an, wer zum angegebenen Zeitpunkt eingeloggt war.

### Beispiel

Auf dem Testrechner hat zuletzt nur der Benutzer `kofler` gearbeitet:

```
user$ last
kofler pts/0 62-47-230-2.adsl Tue Sep 24 13:51 still logged in
kofler pts/0 212-183-46-83.ad Mon Sep 23 19:50 - 20:22 (00:32)
kofler pts/1 91-115-236-11.ad Fri Sep 13 15:39 - 17:51 (02:11)
kofler pts/0 91-115-236-11.ad Fri Sep 13 15:31 - 17:51 (02:20)
```

## lastb [optionen]

lastb zeigt an, welche Login-Versuche zuletzt gescheitert sind. Das Kommando wird mit den gleichen Optionen wie last gesteuert. Es darf nur von root ausgeführt werden. Es wertet die Datei /var/log/btmp aus.

## ldconfig

ldconfig aktualisiert die Links zu allen Bibliotheken und erstellt die Cache-Datei /etc/ld.so.cache, die bei der effizienten Suche nach Bibliotheken hilft. ldconfig wertet die Konfigurationsdatei /etc/ld.so.conf aus. Das Kommando muss nach der manuellen Installation von Bibliotheken ausgeführt werden.

## ldd programm

ldd liefert eine Liste aller Bibliotheken, die zur Ausführung des angegebenen Programms erforderlich sind. Mit dem Kommando stellen Sie auch fest, ob auf dem Rechner alle notwendigen Bibliotheken verfügbar sind.

### Beispiel

Der Gnome-Editor greift unter Fedora 23 auf rund 90 Bibliotheken zurück! Die folgenden Zeilen zeigen die ersten drei davon in alphabetischer Ordnung:

```
root# ldd /usr/bin/gedit | sort
/lib64/ld-linux-x86-64.so.2 (0x00005dea114c000)
libatk-1.0.so.0 => /lib64/libatk-1.0.so.0 (0x00007f7468847000)
libatk-bridge-2.0.so.0 => /lib64/libatk-bridge-2.0.so.0 (0x00007f7465bef000)
...
```

## less [optionen] datei

less zeigt die angegebene Textdatei seitenweise an. Das Kommando wird häufig als Filter verwendet, beispielsweise `ls -l | less` zur seitenweisen Anzeige einer langen Dateiliste.

- ▶ `-m`  
zeigt in der Statuszeile die aktuelle Textposition in Prozent an.
- ▶ `-M`  
zeigt in der Statuszeile den Dateinamen und die Textposition in Zeilen an.
- ▶ `-p suchtext`  
zeigt die erste Zeile an, in der der zu suchende Text gefunden wurde.
- ▶ `-s`  
reduziert mehrere Leerzeilen zu einer Zeile.

Die wichtigsten Kommandos während der Verwendung von `less` sind: `[H]` zur Anzeige eines kurzen Hilfetextes, die Cursortasten zum Bewegen im Text, `[←]`, um eine Zeile nach unten zu scrollen, die Leertaste, um eine Seite nach unten zu scrollen, `[B]`, um eine Seite zurückzuscrollen, `[<]` und `[>]`, um zum Anfang bzw. Ende des Texts zu springen, `[/]` zur Eingabe eines Suchtextes sowie `[Q]` zum Beenden von `less`.

```
lftp [optionen] [site]
```

`lftp` ist ein interaktiver FTP-Client. `lftp` kann aber auch in Scripts eingesetzt werden und durch Kommandos gesteuert werden.

- ▶ `-c "kommando"`  
führt die in einer Zeichenkette angegebenen und durch Strichpunkte getrennten FTP-Kommandos aus (z. B. `lftp -c "open -u user,password server; mirror -R verz"`).
- ▶ `-f datei`  
liest die auszuführenden Kommandos zeilenweise aus einer Datei.

`lftp`-Login-Einstellungen können in der Datei `.netrc` zeilenweise für diverse Server gespeichert werden. Einträge in dieser Datei sehen wie dieses Muster aus:

```
machine backup.hostname.com  login u123456  password x234b0CT2xCb
```

### Beispiel

Das folgende Kommando lädt eine Datei auf einen FTP-Server hoch:

```
root# lftp -c "open -u username,password backupserver; put datei"
```

Um statt einer Datei ein ganzes Verzeichnis zum Backup-Server zu übertragen, verwenden Sie das Kommando `mirror -R`. `mirror` kopiert normalerweise Verzeichnisse

vom FTP-Server auf den lokalen Rechner. `-R` dreht die Übertragungsrichtung um. Auch hierzu ein Beispiel:

```
root# lftp -c "open -u user,passw bserver; mirror -R verzeichnis"
```

```
ln [optionen] quelle [ziel]
ln [optionen] dateien zielverzeichnis
```

`ln` richtet feste oder symbolische Links auf Dateien und Verzeichnisse ein. Die gleiche Funktionalität wie `ln` hat auch das Kommando `cp`, wenn dort die Optionen `-l` oder `-s` angegeben werden.

- ▶ `-b` bzw. `--backup`  
benennt bereits vorhandene gleichnamige Dateien in Backup-Dateien (Name plus `~`-Zeichen) um, anstatt sie zu überschreiben.
- ▶ `-d` bzw. `--directory`  
erzeugt einen festen Link für ein Verzeichnis. Diese Operation ist nur `root` gestattet. Alle anderen Anwender können symbolische Links auf Verzeichnisse erzeugen.
- ▶ `-s` bzw. `--symbolic`  
erzeugt symbolische Links. (Ohne diese Option liefert `ln` feste Links.)

### Beispiel

Das folgende Kommando erzeugt den symbolischen Link `xyz` auf die bereits vorhandene Datei `abc`:

```
user$ ln -s abc xyz
```

```
loadkeys [optionen] filename
```

Das Low-Level-Kommando `loadkeys` lädt eine Tastaturliste für den Textmodus. Die erforderlichen Einstellungsdateien befinden sich üblicherweise im Verzeichnis `/lib/kbd/keymaps`. Wenn Sie einen Dateinamen ohne Pfad angeben, versucht das Kommando selbst, eine geeignete Datei mit der Kennung `.map.gz` zu finden.

Die so veränderten Tastatureinstellungen gelten also nur, wenn Sie in einer Textkonsole arbeiten. Die Einstellungen für den Grafikmodus müssen hingegen im Rahmen der X-Konfiguration verändert werden; bei Distributionen mit `Systemd` können Sie dazu auch das Kommando `localectl` verwenden.

### ► -d

lädt die Default-Tastaturlayout, in der Regel die Datei `/lib/kbd/keymaps/defkeymap.map`.

## Beispiel

Das folgende Kommando aktiviert in den Textkonsolen das deutsche Tastaturlayout:

```
root# loadkeys de-latin1
Lade /lib/kbd/keymaps/i386/qwertz/de.map.gz
```

```
local var[=wert]
```

Das bash-Kommando `local` definiert in Funktionen innerhalb von Shell-Skripts eine lokale Variable. Das Kommando kann nur in selbst definierten Funktionen verwendet werden (siehe [function](#)). Vor und nach dem Gleichheitszeichen dürfen keine Leerzeichen angegeben werden.

```
localectl [optionen] [kommando]
```

Das Kommando `localectl` dient bei Distributionen mit dem Init-System `Systemd` zur Steuerung der Sprach- und Tastatureinstellungen. Es verändert die Dateien `/etc/locale.conf`, `/etc/vconsole.conf` sowie `/etc/X11/xorg.conf/00-keyboard.conf`.

## Optionen

### ► --no-convert

verändert nur die Tastatur für die Konsole (Kommando `set-keymap`) bzw. für den Grafikmodus (Kommando `set-x11-keymap`). Standardmäßig versucht `localectl`, die Tastatureinstellungen sowohl für den Text- als auch für den Grafikmodus zugleich zu verändern. Die betreffenden Dateien haben aber eine unterschiedliche Syntax, weswegen die Konvertierung der Parameter mitunter scheitert.

### ► --no-pager

führt Ausgaben direkt auf der Standardausgabe aus, anstatt `less` zu verwenden.

## Kommandos

### ► list-keymaps

ermittelt eine Liste aller möglichen Tastatur-Layouts für den Textmodus.



► `list-locales`

ermittelt eine Liste aller möglichen Spracheinstellungen.

► `list-x11-keymap-models`  
`list-x11-keymap-layouts`  
`list-x11-keymap-variants` [layout]  
`list-x11-keymap-options`

ermittelt die zulässigen Parameter für die Einstellung des Tastaturlayouts im Grafiksystem X.

► `set-keymap name`

aktiviert das angegebene Tastaturlayout.

► `set-locale name`

aktiviert die angegebene Spracheinstellung.

► `set-x11-keymap layout [modell variante optionen]`

aktiviert das angegebene Tastaturlayout für den Grafikmodus.

► `status`

zeigt die aktuellen Sprach- und Tastatureinstellungen an.

### Beispiel

Die beiden folgenden Kommandos stellen die Sprache Deutsch, den Zeichensatz UTF8 sowie ein deutsches Tastaturlayout ein, wobei die Besonderheiten einer Apple-Tastatur berücksichtigt werden:

```
root# localectl set-locale de_DE.UTF-8
root# localectl set-x11-keymap de de mac grp:alt_shift_toggle
```

### locate muster

locate ermöglicht eine besonders schnelle Suche nach Dateien. Es durchsucht eine zumeist einmal täglich aktualisierte Dateidatenbank, in der das angegebene Muster im vollständigen Dateinamen (inklusive Pfad) vorkommt. Dateien, die nach der letzten Datenbankaktualisierung entstanden sind oder verändert wurden, können aber natürlich nicht gefunden werden.

Zumeist müssen locate und das zugrunde liegende updatedb-System extra installiert (Paket `mlocate`) und oft auch konfiguriert werden.

### **logger** [optionen] nachricht

logger protokolliert eine Nachricht mit dem Syslog-Dienst. Je nach Distribution ist dies zumeist rsyslogd oder bei moderneren Distributionen das Journal (siehe journalctl).

- ▶ `-p n`  
protokolliert die Nachricht in der angegebenen Prioritätsstufe zwischen 0 = emerg und 7 = debug.
- ▶ `-p facility.level`  
protokolliert die Nachricht für den angegebenen Dienst und in der angegebenen Prioritätsstufe. Standardmäßig verwendet logger die Einstellung `user.notice`.
- ▶ `-t tag`  
speichert zusammen mit der Nachricht das angegebene Schlüsselwort. Das vereinfacht eine spätere Suche nach entsprechenden Nachrichten.

### Beispiel

Das folgende logger-Kommando könnte am Ende eines Backup-Scripts stehen. Es protokolliert das erfolgreiche Ende des Backups.

```
user$ logger -p local0.notice -t mybackuptool "Backup completed"
```

### **logname**

logname zeigt den Login-Namen (Benutzernamen) an.

### **logout**

logout oder noch kürzer einfach `[Strg]+[D]` beendet die Sitzung in einer Konsole oder in einem Terminalfenster.

### **lpadmin** [optionen]

lpadmin richtet einen neuen Drucker für das CUPS-Drucksystem ein, verändert dessen Zugriffsrechte für den Netzbetrieb bzw. löscht ihn wieder.

- ▶ `-d name`  
definiert den angegebenen Drucker als Standarddrucker.

- ▶ `-E name`  
aktiviert den angegebenen Drucker.
- ▶ `-p name`  
richtet einen neuen Drucker ein. Mit diversen weiteren Optionen geben Sie die Konfigurationsparameter an (siehe `man lpadmin`).
- ▶ `-x name`  
löscht die Konfiguration für den angegebenen Drucker.

### **lpinfo** [optionen]

lpinfo listet die für CUPS verfügbaren Devices und Treiber auf.

- ▶ `-l`  
liefert besonders ausführliche Informationen. Die Option muss mit `-m` oder `-v` kombiniert werden.
- ▶ `-m`  
liefert eine Liste aller verfügbaren Druckertreiber.
- ▶ `-v`  
liefert eine Liste aller bekannten Druck-Devices.

### **lpoptions** [optionen]

lpoptions zeigt die Optionen von CUPS-Druckern an bzw. verändert sie.

- ▶ `-l`  
liefert eine Liste der verfügbaren Optionen und deren aktueller Einstellung.
- ▶ `-o optionsname=wert`  
verändert die Einstellung der angegebenen Option.
- ▶ `-p name`  
gibt den gewünschten Drucker an. (Ohne die Option bezieht sich `lpoptions` auf den Standarddrucker.)

### **lpq** [optionen]

lpq liefert eine Liste aller zwischengespeicherten Dateien bzw. Druckaufträge. Dabei werden auch die Größe der Datei sowie eine Jobnummer angegeben. Diese Jobnummer können Sie als Parameter von **lprm** angeben, um eine Datei aus dem Drucker-Spooler zu entfernen.

- ▶ **-a**  
zeigt die Druckjobs aller Warteschlangen an.
- ▶ **-Pname**  
zeigt die Druckjobs der angegebenen Warteschlange *name* an.

### **lpr** datei

lpr druckt die angegebene Datei aus.

- ▶ **-l**  
umgeht das Filtersystem und sendet die Druckerdaten unverändert an den Drucker. Die Option ist dann sinnvoll, wenn eine Druckdatei bereits im druckerspezifischen Format vorliegt.
- ▶ **-o options**  
übergibt diverse Zusatzparameter, z.B. **-o media=A4** oder **-o page-ranges=23-27, 29,31**. Zahlreiche Beispiele finden Sie in der CUPS-Dokumentation:  
<http://www.cups.org/documentation.php/options.html>
- ▶ **-Pname**  
verwendet die Warteschlange *name* statt des Standarddruckers. Beachten Sie, dass der Option kein Leerzeichen folgt!

### **lprm** [optionen] [id]

lprm bricht den aktuellen bzw. den durch die ID-Nummer angegebenen Druckjob ab.

- ▶ **-Pname**  
gibt die Warteschlange an.

**lpstat** [optionen]

lpstat zeigt Informationen über CUPS-Klassen, -Drucker und deren Druckjobs an. lpstat funktioniert gleichermaßen für lokale Drucker wie für Netzwerkdrucker.

- ▶ -a  
zeigt für alle Drucker an, ob sie bereit sind, Druckaufträge entgegenzunehmen.
- ▶ -c  
zeigt alle Klassen an.
- ▶ -d  
zeigt den Standarddrucker an.
- ▶ -s  
zeigt eine Statusübersicht an (Standarddrucker, Liste aller Klassen und Drucker etc.).
- ▶ -t  
zeigt alle verfügbaren Informationen an.
- ▶ -v  
zeigt alle Drucker an.

**ls** [-optionen] [pfad]

ls zeigt eine Liste aller Dateien und Verzeichnisse an. Wenn ls ohne weitere Parameter oder Optionen verwendet wird, liefert das Kommando eine mehrspaltige, nach Dateinamen sortierte Tabelle, in der alle Dateien, Links und Verzeichnisse im aktuellen Verzeichnis angezeigt werden.

- ▶ -a bzw. -all  
zeigt auch Dateien an, die mit . beginnen. Eine ganz ähnliche Wirkung hat die Option -A. Der einzige Unterschied besteht darin, dass die Dateien . und .. (Verweis auf das aktuelle und das übergeordnete Verzeichnis) nicht angezeigt werden.
- ▶ --color [=never|auto|always]  
verwendet unterschiedliche Farben für unterschiedliche Dateitypen (Links, Verzeichnisse etc.) bzw. deaktiviert die Farbdarstellung. Mit dircolors können Sie die Farben selbst konfigurieren.

► `-d` bzw. `--directory`

zeigt nur den Namen des Verzeichnisses, nicht aber seinen Inhalt an. Die Option ist beispielsweise dann sinnvoll, wenn als Pfad ein Verzeichnisname angegeben wird und die Zugriffsrechte dieses Verzeichnisses überprüft werden sollen (und nicht sein Inhalt).

► `-h`

zeigt die Dateigrößen in kByte, MByte und GByte an (*human readable*).

► `-i` bzw. `--inode`

zeigt zusätzlich zu den restlichen Informationen auch den I-Node der Datei an. (Der I-Node ist eine interne Identifikationsnummer der Datei, die zur Linux-internen Dateiverwaltung benötigt wird.) Die Option kann zur Erkennung von Links eingesetzt werden. (Durch Links verknüpfte Dateien haben denselben I-Node.)

► `-I datei` bzw. `--ignore datei-Imuster`

schließt die angegebenen Dateien von der Anzeige aus. `-I*ps` verhindert beispielsweise, dass Dateien mit der Endung `ps` angezeigt werden. Wenn nach `-I` nicht eine einzelne Datei, sondern ein Dateimuster angegeben wird, dann darf zwischen `-I` und dem Muster kein Leerzeichen eingegeben werden!

► `-l` bzw. `--format=long` oder `--format=verbose`

zeigt zusätzlich zum Dateinamen weitere Informationen an: die Dateigröße in Bytes, die Zugriffsrechte etc. Zur Anzeige jeder Datei wird eine eigene Zeile verwendet (statt der platzsparenden mehrspaltigen Auflistung).

► `-L` bzw. `--dereference`

zeigt bei einem symbolischen Link auf ein Verzeichnis nicht den Pfad des Links, sondern den Inhalt des Ursprungsverzeichnisses an.

► `-o` bzw. `--no-color`

verzichtet auf unterschiedliche Farben oder Schriftarten.

► `-p` bzw. `-F`

hängt an die Dateinamen ein Sonderzeichen an, das den Typ der Datei kennzeichnet. Diese Option wird bei manchen Linux-Distributionen in `/etc/profile` durch eine alias-Abkürzung standardmäßig aktiviert. Die wichtigsten Sonderzeichen sind: `/` für Verzeichnisse, `@` für symbolische Links, `*` für ausführbare Dateien und `|` für FIFOs.

- ▶ `-r` bzw. `--reverse`  
dreht die Sortierreihenfolge um. Die Option wird oft in Kombination mit `-t` oder `-S` verwendet.
- ▶ `-R` bzw. `--recursive`  
erfasst auch Dateien in Unterverzeichnissen.
- ▶ `-S` bzw. `--sort=size`  
sortiert die Dateien nach ihrer Größe (die größte Datei zuerst).
- ▶ `-t` bzw. `--sort=time`  
sortiert die Dateien nach Datum und Uhrzeit der letzten Änderung (die neueste Datei zuerst).
- ▶ `-u` bzw. `--sort=access`  
sortiert die Dateien nach Datum und Uhrzeit des letzten Lesezugriffs. Die Option muss zusammen mit `-t` angegeben werden (sonst sortiert `ls` überhaupt nicht).
- ▶ `-X` bzw. `--sort=extension`  
sortiert die Dateien nach ihrer Kennung (d. h. nach der Buchstabenkombination, die auf den letzten `.` im Dateinamen folgt).
- ▶ `-Z` bzw. `--context`  
zeigt die SELinux-Kontextinformationen. Mehr Informationen liefert `--lcontext`, weniger Informationen gibt `--scontext`.

`ls` zeigt leider nicht den gesamten Speicherbedarf aller aufgelisteten Daten an. Diese Aufgabe übernimmt das Kommando `du`. `ls` ist nicht in der Lage, die Zugriffsbits einer Datei oktal anzuzeigen. Dazu verwenden Sie das Kommando `stat`.

## Beispiel

Das folgende Kommando zeigt alle Dateien im aktuellen Verzeichnis an und sortiert sie nach dem Datum (die neueste Datei zuletzt).

```
user$ ls -ltr
-rw-r--r-- 1 kofler kofler 1681276 2014-03-02 10:01 cimg3079.jpg
-rw-r--r-- 1 kofler kofler 1582496 2014-03-02 10:01 cimg3014.jpg
-rw-r--r-- 1 kofler kofler 1615070 2014-03-02 10:01 cimg2965.jpg
...
```

Kurz einige Anmerkungen zur Interpretation des `ls`-Ergebnisses: Die zehn Zeichen am Beginn der Zeile geben den Dateityp und die Zugriffsbits an. Als Dateityp kommen

infrage: der Bindestrich - für eine normale Datei, d für ein Verzeichnis (*directory*), b oder c für eine Device-Datei (*block* oder *char*) oder l für einen symbolischen Link.

Die nächsten drei Zeichen (*rwX*) geben an, ob der Besitzer die Datei lesen, schreiben und ausführen darf. Analoge Informationen folgen für die Mitglieder der Gruppe sowie für alle anderen Systembenutzer.

Die Zahl im Anschluss an die zehn Typ- und Zugriffszeichen gibt an, wie viele Hard-Links auf die Datei verweisen. Die weiteren Spalten geben den Besitzer und die Gruppe der Datei an (hier jeweils *kofler*), die Größe der Datei, Datum und Uhrzeit der letzten Änderung und zuletzt den Dateinamen.

**lsattr** [optionen] dateien

lsattr zeigt den Zustand der Zusatzattribute von Dateien oder Verzeichnissen in Linux-Dateisystemen an (ext2 bis ext4, btrfs, xfs etc.). Diese Zusatzattribute steuern beispielsweise, ob Dateien komprimiert werden sollen, ob Änderungen an Dateien mit dem Copy-on-Write-Verfahren durchgeführt werden sollen etc. Es hängt von der Implementierung des jeweiligen Dateisystems ab, welche Attribute tatsächlich beachtet werden – bei den meisten Dateisystemen nur ganz wenige (siehe auch die Beschreibung des Kommandos *chattr* zur Veränderung dieser Attribute).

- ▶ -a  
zeigt auch versteckte Dateien an.
- ▶ -d  
zeigt die Attribute von Verzeichnissen an, nicht die Attribute der Dateien in den Verzeichnissen.
- ▶ -R  
zeigt rekursiv die Attribute aller Dateien im Verzeichnisbaum an.

### Beispiel

Auf dem Testsystem (eine openSUSE-Leap-Installation mit btrfs-Dateisystem) ist für die Verzeichnisse */var/lib/mysql* und */var/lib/libvirt/images* das Attribut *C* gesetzt. Es deaktiviert die Copy-on-Write-Funktion für Dateien in diesem Verzeichnis.

```
root# lsattr -d /var /var/lib /var/lib/libvirt/images /var/lib/mariadb
----- /var
----- /var/lib/
-----C /var/lib/libvirt/images
-----C /var/lib/mariadb/
```



```
lsblk [optionen] [iodevice]
```

lsblk liefert eine hierarchische Liste sämtlicher Block-Devices bzw. aller Devices, die sich auf einem Datenträger befinden. Das Kommando ist ausgezeichnet geeignet, um rasch einen guten Überblick über alle Partitionen und Logical Volumes zu erhalten.

- ▶ -a  
bezieht auch leere Devices in das Ergebnis ein.
- ▶ -b  
gibt Größenangaben in Byte an.
- ▶ -f  
zeigt zu jedem Device auch den Dateisystemtyp, den Label und die UUID an.
- ▶ -p  
gibt vollständige Device-Namen an, also z. B. dev/mapper/fedora-swap bei Logical Volumes.

### Beispiel

Das folgende lsblk-Ergebnis ist auf einem Fedora-Rechner mit zwei Festplatten und einem DVD-Laufwerk entstanden. Das root-Dateisystem und die Swap-Partition befinden sich in Logical Volumes.

```
root# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda     8:0    0 232,9G  0 disk
  sda1   8:1    0  487M  0 part /boot/efi
  sda2   8:2    0  14,8G  0 part
    fedora-swap 253:0    0  500M  0 lvm  [SWAP]
    fedora-root 253:1    0 14,3G  0 lvm  /
sda3    8:3    0  93,2G  0 part /virt
sdb     8:16   0 931,5G  0 disk
  sdb1   8:17   0  128M  0 part
  sdb2   8:18   0 146,4G  0 part
  sdb3   8:19   0 783,5G  0 part /data
sr0     11:0   1 1024M  0 rom
```

```
lscpu [optionen]
```

lscpu zeigt Detailinformationen zur CPU, deren Cores und Cache-Speicher sowie gegebenenfalls zum Virtualisierungssystem an.

- ▶ `-e`  
zeigt nur eine Zusammenfassung der Eckdaten in einer gut lesbaren Form an.
- ▶ `-p`  
wie `-e`, liefert die Daten aber in einer Form, die einfach in Scripts verarbeitet werden kann.

### `lshw` [optionen]

Das Kommando `lshw` aus dem gleichnamigen Paket erstellt eine hierarchisch geordnete Liste aller Hardware-Komponenten des Rechners.

- ▶ `-businfo`  
listet die Komponenten zeilenweise auf und gibt zu jeder Komponente eine ID-Nummer für den verwendeten SCSI-, USB- oder PCI-Bus an.
- ▶ `-html`  
formatiert die Ausgabe als HTML-Dokument.
- ▶ `-short`  
liefert nur eine kompakte Zusammenfassung.
- ▶ `-xml`  
formatiert die Ausgabe als XML-Dokument.

### Beispiel

```
root# lshw -businfo
```

Bus info	Device	Class	Description
		system	Raspberry Pi 2 Model B Rev 1.1
		bus	Motherboard
cpu@0		processor	cpu
cpu@1		processor	cpu
cpu@2		processor	cpu
cpu@3		processor	cpu
		memory	925MiB System memory
usb@1	usb1	bus	DWC OTG Controller
usb@1:1		bus	USB hub
usb@1:1.1		generic	SMSC9512/9514 Fast Ethernet Adapter
...			

**lsmod**

lsmod liefert eine Liste aller Module, die momentan in den Kernel geladen sind.

**lsof** [optionen] [datei/schnittstelle]

Ohne weiteren Parameter liefert lsof eine zumeist sehr lange Liste aller momentan offenen Dateien bzw. Schnittstellen und der ihnen zugeordneten Prozesse. lsof datei liefert Informationen zum Prozess, der diese Datei offen hält.

▶ *-i adresse*

liefert Informationen zu Prozessen, die die angegebene Netzwerkadresse nutzen. Die Adresse setzt sich in der Form [46][protocol][@hostname|hostaddr][:service|port] zusammen. Beispielsweise liefert *-i 4tcp* Informationen zu allen Prozessen, die das Protokoll TCP in der IP-Version 4 nutzen.

▶ *-n*

verzichtet auf die Auflösung von Netzwerknamen.

▶ *-N*

berücksichtigt auch NFS-Dateien.

▶ *-u user*

liefert nur Informationen zu Dateien/Schnittstellen, die vom angegebenen Benutzer genutzt werden. Es können auch mehrere Benutzer angegeben werden (Name oder UID, getrennt durch Kommas).

▶ *-X*

ignoriert alle offenen TCP- und UDP-Dateien.

**Beispiele**

Die beiden folgenden Kommandos zeigen alle Prozesse, die das Protokoll UDP bzw. den Port 22 nutzen:

```
root# lsof -i udp
ntpd      3696      ntp    16u  IPv4  9026          UDP *:ntp
portmap   4745  daemon    3u  IPv4 12931          UDP *:sunrpc
rpc.statd 4764   statd    5u  IPv4 12962          UDP *:700
...
```

```
root# lsof -i :22
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
sshd	5559	root	3u	IPv6	14097		TCP	*:ssh (LISTEN)
sshd	7729	root	3r	IPv6	33146		TCP	mars.sol:ssh->merkur.sol:45368 (ESTABLISHED)

### lspci [optionen]

lspci liefert Informationen über den PCI-Bus und sämtliche daran angeschlossenen Geräte.

► -tv

liefert eine baumförmige Geräteliste, die deutlich macht, wie die Geräte miteinander verbunden sind.

► -v bzw. -vv bzw. -vvv

liefert noch mehr Details.

### lsscsi [optionen]

lsscsi liefert Informationen über alle angeschlossenen SCSI- und SATA-Geräte.

► -C

zeigt die Daten im selben Format wie `/proc/scsi/scsi` an.

► -H

liefert eine Liste der SCSI-Hosts (statt der SCSI-Geräte).

► -l

liefert ausführliche Informationen zu jedem Gerät.

### Beispiel

Das folgende Ergebnis ist auf einem Rechner mit einer SSD, einer herkömmlichen Festplatte und einem DVD-Laufwerk entstanden. Alle drei Geräte sind mit SATA verkabelt.

```
root# lsscsi
```

[0:0:0:0]	disk	ATA	Samsung SSD 840	6B0Q	/dev/sda
[0:0:1:0]	cd/dvd	Optiarc	DVD RW AD-7240S	1.02	/dev/sr0
[1:0:1:0]	disk	ATA	WDC WD10EADS-00L	1A01	/dev/sdb

**lsusb** [optionen]

lsusb liefert Informationen über alle angeschlossenen USB-Geräte.

## ► -t

rückt die Ausgabe baumartig ein und verdeutlicht so, an welchem Bus welches Gerät angeschlossen ist.

## ► -v

liefert ausführliche Informationen zu jedem Gerät.

**Beispiel**

Das folgende Kommando zeigt, dass an den Computer ein USB-WLAN-Adapter angesteckt ist:

```
user$ lsusb
```

```
...
```

```
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co.,  
    Ltd EW-7811Un 802.11n Wireless Adapter [Realtek RTL8188CUS]
```

**luksformat** [optionen] device

luksformat richtet mit cryptsetup luksFormat einen Crypto-Container ein, aktiviert diesen mit cryptsetup luksOpen und richtet auf dem resultierenden Device ein VFAT-Dateisystem ein. Wenn Sie ein anderes Dateisystem wünschen, verwenden Sie beispielsweise -t ext4.

**lvcreate** [optionen] name

Das LVM-Kommando lvcreate erzeugt ein neues Logical Volume (LV) innerhalb der durch name angegebenen Volume Group (VG).

► -i *n*

verteilt das LV gleichmäßig über die angegebene Anzahl von PVs (Physical Volumes). Das setzt voraus, dass die VG aus zumindest so vielen PVs besteht. Sofern sich die PVs auf unterschiedlichen Festplatten befinden, erzielen Sie damit einen ähnlichen Effekt wie mit RAID-0 (Striping).

► `-L size`

gibt die gewünschte Größe an. Die Größenangabe erfolgt standardmäßig in MByte, mit dem entsprechenden Suffix aber auch in Byte, kByte, GByte oder TByte (z. B. `-L 2G`).

► `-n lvname`

gibt den LV-Namen an. Wenn diese Option fehlt, verwendet das Kommando den Namen `lvoldn`.

► `-s`

erzeugt einen Snapshot, also eine statische Kopie eines vorhandenen LVs. `name` gibt in diesem Fall nicht die VG, sondern das zugrunde liegende LV an. Die Größenangabe `-L` gibt den Speicher an, der für kopierte Sektoren benötigt wird, wenn sich das zugrunde liegende LV ändert. Ist dieser Speicher erschöpft, endet die Lebensdauer des Snapshots.

### Beispiel

Das folgende Kommando richtet in der VG `myvg1` ein neues Logical Volume mit dem Namen `myvol1` in der Größe von 50 GByte ein:

```
root# lvcreate -L 50G -n myvol1 myvg1
Logical volume "myvol1" created
```

Durch das Kommando wird gleichzeitig auch die Datei `/dev/myvg1/myvol2` erzeugt. Dabei handelt es sich um einen Link auf die Datei `/dev/mapper/myvg1-myvol2`. Das LV kann jetzt unter einem dieser beiden Device-Namen wie eine gewöhnliche Festplattenpartition verwendet werden. Das folgende Kommando richtet dort ein Dateisystem ein:

```
root# mkfs.ext4 /dev/myvg1/myvol1
```

```
lvdisplay [optionen] lvname
```

`lvdisplay` zeigt Detailinformationen zu einem LV an.

```
lvextend [optionen] lvname [pvname]
```

`lvextend` vergrößert den Speicher für das angegebene LV-Device. Den neuen Gesamtspeicher geben Sie mit der Option `-L` an (siehe [lvcreate](#)). Alternativ können Sie die Schreibweise `-L +size` verwenden, um die gewünschte Änderung anzugeben.

Der benötigte Speicher wird innerhalb derselben VG reserviert, in der sich das LV befindet. Wenn die VG aus mehreren Physical Volumes (PVs) zusammengesetzt ist, können Sie durch die optionale Angabe von `pvname` auswählen, welche PV den Speicher zur Verfügung stellt.

Ein Beispiel zur Anwendung von `lvextend` finden Sie bei der Beschreibung des Kommandos `resize2fs` zur Veränderung der Größe von ext-Dateisystemen.

### lv`m` [kommando]

`lvm` ist das zentrale Administrationskommando für den Logical Volume Manager. `lvm` befindet sich üblicherweise im Paket `lvm2` und muss bei vielen Distributionen extra installiert werden, wenn nicht bereits bei der Installation ein LVM-System eingerichtet wurde.

Wenn Sie `lvm` ohne weitere Kommandos ausführen, gelangen Sie in eine Shell, in der Sie interaktiv LVM-Kommandos ausführen können. `help` liefert eine Liste aller zur Auswahl stehenden Kommandos. Alternativ können Sie an `lvm` ein Kommando übergeben, das dann sofort ausgeführt wird.

Alle LVM-Kommandos können wahlweise via `lvm` oder direkt ausgeführt werden. Die Kommandos `lvm lvcreate xxx` und `lvcreate xxx` sind daher gleichwertig.

Aus Gründen der Übersichtlichkeit habe ich mich in diesem Buch dazu entschlossen, die wichtigsten LVM-Kommandos separat zu beschreiben. Kommandos zur Bearbeitung von Logical Volumes beginnen mit den Anfangsbuchstaben `lv`, solche zur Administration von Physical Volumes beginnen mit `pv`, und Kommandos zum Verwalten von Volume Groups beginnen mit `vg`.

### lvreduce [optionen] lvname

`lvreduce` verkleinert den Speicherplatz des angegebenen LV-Device. Wenn das LV ein Dateisystem enthält, muss dieses *vorher* verkleinert werden, andernfalls kommt es zu Datenverlusten! Sie können wahlweise die neue Gesamtgröße in der Form `-L size` angeben oder die gewünschte Änderung in der Form `-L -size`.

### lvremove [optionen] lvname

`lvremove` löscht das angegebene LV.

► `-f`

verzichtet auf Rückfragen.

```
lvrename oldlvname newlvname
```

lvrename gibt dem LV einen neuen Namen.

```
lvscan [optionen]
```

lvscan listet alle LVs auf.

```
lzop [optionen] [dateiname]
```

lzop aus dem gleichnamigen Paket komprimiert bzw. dekomprimiert Dateien. lzop erreicht deutlich schlechtere Kompressionsraten als gzip, bzip2 oder xz, ist dafür aber *viel* schneller. Die Syntax von lzop orientiert sich am gzip-Kommando.

lzop ist das perfekte Kommando, wenn Sie Dateien platzsparend sichern möchten (z. B. für ein Backup), dafür aber nicht allzu viel CPU-Leistung investieren möchten. Dabei werden Sie lzop zumeist im Pipe-Modus verwenden, wie bei den beiden tar-Beispielen unten. In den meisten Fällen sparen Sie durch das Komprimieren nicht nur Platz, sondern auch Zeit! Bei vielen Dateien ist die Zeitersparnis bei der Datenspeicherung (I/O) bzw. bei der Netzwerkübertragung größer als die zum Komprimieren erforderliche Zeit.

- ▶ `-c` bzw. `--stdout` oder `--to-stdout`  
leitet das Ergebnis auf die Standardausgabe um.
- ▶ `-d` bzw. `--decompress` oder `--uncompress`  
dekomprimiert die angegebene Datei, anstatt sie zu komprimieren.
- ▶ `-n`  
steuert die Geschwindigkeit und Qualität der Kompression. -1 ist am schnellsten, -9 liefert die kleinsten Dateien. Standardmäßig gilt -3.
- ▶ `-U`  
überschreibt die vorhandene Datei. (Standardmäßig bleibt die ursprüngliche Datei erhalten. Die neue, komprimierte Datei erhält das Suffix `.lzo`.)

### Beispiele

Das erste Kommando komprimiert die Datei und liefert `datei.lzo`. Das zweite Kommando dekomprimiert die Datei wieder:



```
user$ lzop datei
user$ lzop -d datei.lzw
```

Die beiden folgenden Beispiele zeigen die Anwendung von `lzop` in Kombination mit `tar`:

```
user$ tar cf -c verz | lzop -c > backup.tar.lzo    (Archiv erstellen)
user$ lzop -d < backup.tar.lzo | tar xf -          (Archiv auspacken)
```

```
mail [optionen] mail@hostname [< nachrichtentext]
```

Mit dem Kommando `mail` aus dem Paket `mailx` oder `mailutils` (Debian, Ubuntu) können Sie lokale, in `/var/mail/name` gespeicherte E-Mails lesen. Die Bedienung ist aber gelinde gesagt gewöhnungsbedürftig. Wenn Sie einen Mail-Client für den Textmodus suchen, installieren Sie besser `mutt`.

Die Stärke von `mail` liegt vielmehr darin, dass Sie damit unkompliziert E-Mails an einen lokalen Mail-Server übergeben können, wahlweise interaktiv oder in Scripts. Interaktiv führen Sie einfach `mail empfänger@site` aus. Sie können nun weitere Empfänger (CC-Adressen), das Subject und schließlich den Nachrichtentext angeben, den Sie mit `[Strg]+[D]` abschließen. Damit wird die E-Mail an den lokalen Mail-Server übergeben, wobei als Absender `loginname@hostname` verwendet wird.

Für den Script-gesteuerten Mail-Versand müssen Sie sich mit den `mail`-Optionen anfreunden, von denen hier nur die wichtigsten drei vorgestellt werden:

► `-a "header"`

gibt eine zusätzliche Header-Zeile an. Das kann dazu verwendet werden, um eine BCC-Adresse hinzuzufügen oder um den Zeichensatz der Nachricht einzustellen (siehe das Beispiel).

► `-A datei`

fügt der E-Mail eine Datei hinzu.

► `-s "subject"`

gibt den Text für die Betreffzeile an.

### Beispiel

Das folgende Kommando sendet den UTF-8-Text aus `nachricht.txt` an die angegebenen Empfänger:

```
user$ mail name1@firma-abc.de,name1@firma-abc.de \  
-s "Subject 1 2 3 äöü" \  
-a "BCC: blindcopy@firma-abc.de" \  
-a "Content-Type: text/plain; charset=UTF-8" < nachricht.txt
```

### **makepasswd** [optionen]

`makepasswd` aus dem gleichnamigen Paket erzeugt ein neues, zufälliges Passwort. Das Kommando ist gut geeignet, wenn Sie für einen neuen Account ein Passwort benötigen, aber nicht Ihre eigene Fantasie strapazieren möchten.

Unter CentOS, Fedora und RHEL steht `makepasswd` nicht zur Verfügung. Dort installieren Sie stattdessen das Paket `expect`, das das Kommando `mkpasswd` enthält. Unter (open)SUSE steht auch `mkpasswd` nicht zur Verfügung. Dort können Sie auf [pwgen](#) ausweichen.

- ▶ `--chars n`  
gibt die gewünschte Passwortlänge an (standardmäßig zehn Zeichen).
- ▶ `--count n`  
liefert *n* Passwörter (standardmäßig nur eines).
- ▶ `--strings zeichenkette`  
verwendet die in der Zeichenkette enthaltenen Zeichen, um das Passwort zusammenzusetzen. Standardmäßig verwendet das Kommando nur Groß- und Kleinbuchstaben (ASCII) und Ziffern, aber keine Sonderzeichen.

### Beispiel

Das folgende Kommando liefert ein Passwort mit zwölf Zeichen:

```
user$ makepasswd --chars 12  
TaQKD1VGYmax
```

### **man** [gruppe] [optionen] name

`man` zeigt Online-Informationen zum angegebenen Kommando bzw. zur angegebenen Datei an. Durch die Angabe einer Gruppe kann die Suche eingeschränkt werden. Wichtige Gruppen sind 1 (Benutzerkommandos), 5 (Konfigurationsdateien) und 8 (Kommandos zur Systemadministration). Bei den meisten Distributionen werden die `man`-Texte durch `less` angezeigt, d. h., es gelten die beim Kommando `less` in [Tabelle 17](#) zusammengefassten Tastenkürzel zum Blättern bzw. Suchen im Hilfetext.

## ► -a

zeigt der Reihe nach alle gleichnamigen `man`-Seiten an. Ohne diese Option wird gewöhnlich nur die erste von mehreren gleichnamigen Dateien aus unterschiedlichen Themengebieten angezeigt. Bei manchen Linux-Distributionen gilt `-a` als Standardeinstellung.

► -f *schlüsselwort*

zeigt die Bedeutung eines Schlüsselworts an (einen einzeiligen Text). Mit dieser Option entspricht `man` dem Kommando `whatis thema`.

► -k *schlüsselwort*

zeigt eine Liste aller vorhandenen `man`-Texte an, in denen das Schlüsselwort vorkommt. Dabei wird allerdings keine Volltextsuche durchgeführt. Vielmehr werden nur die Schlüsselwörter jedes `man`-Textes analysiert. Mit dieser Option entspricht `man` dem Kommando `apropos thema`.

**md5sum** dateien

`md5sum` berechnet Prüfsummen zu allen angegebenen Dateien. Prüfsummen des *Message-Digest Algorithm 5* werden oft dazu verwendet, um nach der Übertragung einer Datei sicherzustellen, dass die Datei unverändert ist.

**mdadm** [optionen]

Das Kommando `mdadm` aus dem gleichnamigen Paket hilft bei der Verwaltung von Software-RAID-Verbunden. Das Kommando wird durch unzählige Optionen gesteuert. Dabei gibt zumeist eine Option den Modus an (z.B. `--create` bzw. `-C`, um einen neuen RAID-Verbund einzurichten). Die weiteren Optionen steuern Details der gewünschten Operation.

► *mddevice --add device*

fügt dem Verbund eine Partition hinzu.

► *--assemble mddevice device1 device2 ... device*m**

fügt den Verbund `mddevice` aus den Festplattenpartitionen `device1` bis `devicem` neu zusammen. Das funktioniert nur, wenn `device1` bis `devicem` vorher mit `--create` als Bestandteile eines RAID-Verbunds initialisiert wurden.

- ▶ `--assemble --scan mddevice`  
setzt die RAID-Partition *mddevice* aus den in *mdadm.conf* angegebenen Festplattenpartitionen zusammen.
- ▶ `--create mddevice --level=n --raid-devices=m device1 device2 ... devicem`  
erzeugt den neuen Verbund *mddevice* für den RAID-Level *n*, der aus den *m* Festplattenpartitionen *device1* bis *device*m** besteht.
- ▶ `--detail mddevice`  
liefert Detailinformationen über die RAID-Partition.
- ▶ `--examine device`  
liefert Detailinformationen über die angegebene Festplattenpartition, die Bestandteil eines RAID-Verbunds sein muss.
- ▶ `--examine --scan`  
liefert eine Zusammenfassung über alle verfügbaren RAID-Partitionen in der Syntax von *mdadm.conf*.
- ▶ `mddevice --fail device`  
markiert die Festplattenpartition *device* als fehlerhaft und deaktiviert sie so.
- ▶ `--grow mddevice --size=max`  
vergrößert den Verbund, sodass die zugrunde liegenden Festplattenpartitionen optimal genutzt werden. Das Kommando ist in dieser Form nur nach einer Vergrößerung der Festplattenpartitionen sinnvoll.
- ▶ `--grow mddevice --raid-devices=n`  
vergrößert die Anzahl der in den Verbund integrierten Festplattenpartitionen (nur für die RAID-Level 1, 5 und 6).
- ▶ `--monitor device`  
überwacht die aktiven RAID-Verbunde. Das Kommando wird üblicherweise beim Hochfahren des Rechners gestartet und sendet bei Problemen eine Mail an den Administrator (Option *-m*).
- ▶ `--query device`  
testet, ob die Partition eine RAID-Partition bzw. Bestandteil eines RAID-Verbunds ist.

► `mddevice --remove device`

entfernt die Festplattenpartition *device* aus dem RAID-Verbund. *device* darf nicht aktiv sein und muss gegebenenfalls vorher mit `--fail` deaktiviert werden.

► `--stop mddevice`

deaktiviert den angegebenen Verbund. Der Verbund kann nun mit `--assemble` neu zusammengesetzt werden.

► `--zero-superblock device`

löscht die RAID-Metadaten einer Festplattenpartition.

## Beispiele

Die beiden folgenden Kommandos bilden aus den beiden Partitionen `/dev/sda3` und `/dev/sdb3` einen RAID-0-Verbund und richten dort ein `ext4`-Dateisystem ein. Die Partitionen müssen zuvor als RAID-Partitionen gekennzeichnet werden. Wenn Sie zum Partitionieren `fdisk` verwenden, stellen Sie die Partitions-ID-Nummer mit dem Kommando `[T]` auf den hexadezimalen Wert `fd`. Bei `parted` führen Sie `set partititionsnummer raid on` aus.

```
root# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sda3 /dev/sdb3
mdadm: array /dev/md0 started.
root# mkfs.ext4 /dev/md0
```

Die Vorgehensweise beim Einrichten eines RAID-1-Verbunds ist exakt dieselbe wie bei RAID-0. Einzig das Kommando zum Einrichten des RAID-Systems sieht ein wenig anders aus und enthält nun `--level=1` statt `--level=0`:

```
root# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda3 /dev/sdb3
mdadm: array /dev/md0 started.
root# mkfs.ext4 /dev/md0
```

Um die Funktionsweise eines RAID-1-Verbunds zu testen, markieren Sie eine Partition als defekt:

```
root# mdadm /dev/md0 --fail /dev/sdb3
```

Sie können den Verbund weiter nutzen; alle Änderungen werden nun aber nur noch auf der verbleibenden Festplattenpartition gespeichert. Um `/dev/sdb3` wieder zu `/dev/md0` hinzuzufügen, müssen Sie die als defekt gekennzeichnete Partition zuerst explizit entfernen:

```
root# mdadm --remove /dev/md0 /dev/sdb3
root# mdadm --add /dev/md0 /dev/sdb3
```

Es beginnt nun die automatische Resynchronisation der beiden Partitionen, die je nach der Größe des Verbunds geraume Zeit dauert (Richtwert: ca. 20 Minuten pro

100 GByte bei herkömmlichen SATA-Festplatten). Immerhin können Sie in dieser Zeit weiterarbeiten. Das Dateisystem wird allerdings langsamer als sonst reagieren. `cat /proc/mdstat` gibt Auskunft darüber, wie weit der Synchronisationsprozess fortgeschritten ist.

Um alle durchgeführten Änderungen in der RAID-Konfigurationsdatei `mdadm.conf` zu speichern, führen Sie das folgende Kommando aus. Anschließend müssen Sie mit einem Editor eventuell schon früher eingetragene RAID-Verbunde aus der Konfigurationsdatei entfernen, damit die Datei frei von Doppelgängern ist.

```
root# mdadm --examine --scan >> /etc/mdadm/mdadm.conf
```

### **mkdir** verzeichnis

`mkdir` erstellt ein neues Verzeichnis. Die beiden wichtigsten Optionen lauten:

- ▶ `-m` *modus* bzw. `--mode=modus`  
setzt die Zugriffsrechte des neuen Verzeichnisses wie durch *modus* angegeben (siehe `chmod`).
- ▶ `-p` bzw. `--parents`  
erstellt auch Zwischenverzeichnisse. Wenn Sie `mkdir a/b/c` ausführen und die Verzeichnisse `a` sowie `a/b` noch nicht existieren, werden auch diese Verzeichnisse erstellt.

### **mkfifo** datei

`mkfifo` richtet eine FIFO-Datei ein (*First In, First Out*). FIFO-Dateien funktionieren im Prinzip wie Pipes und ermöglichen den Datenaustausch zwischen zwei Programmen.

#### Beispiel

Im folgenden Beispiel richtet `mkfifo` eine FIFO-Datei ein. `ls` leitet das Inhaltsverzeichnis auf diese Datei um. `more` liest es von dort. `ls` muss mit `&` als Hintergrundprozess gestartet werden, weil der Prozess `ls` erst dann zu Ende ist, wenn `more` alle Daten aus `fifo` gelesen hat. Die Anzeige des Inhaltsverzeichnisses via `more` könnte natürlich viel einfacher auch ohne eine FIFO-Datei bewerkstelligt werden, nämlich durch `ls -l | more`.

```
user$ mkfifo fifo
user$ ls -l > fifo &
user$ more < fifo
```

```
mkfs [optionen] device [blocks]
```

mkfs richtet auf einer zuvor partitionierten Festplatte ein Dateisystem ein. mkfs kann nur von root ausgeführt werden. Das Programm verzweigt je nach angegebenem Dateisystem in das Kommando `mkfs.dateisystemtyp`. Die wichtigsten dateisystem-spezifischen `mkfs.xxx`-Kommandos werden im Folgenden beschrieben.

► `-t dateisystemtyp`

gibt den Typ des Dateisystems an. Infrage kommen z. B. `ext4` oder `xfs`. Die Option `-t` muss als erste Option angegeben werden! Alle weiteren Optionen werden an das Kommando weitergegeben, das das Dateisystem tatsächlich einrichtet. Sie sind vom Typ des Dateisystems abhängig.

```
mkfs.btrfs [optionen] device1 [device2 device3 ...]
```

mkfs.btrfs richtet ein btrfs-Dateisystem ein. Das Kommando ist je nach Distribution Teil des Pakets `btrfs-tools`, `btrfs-progs` oder `btrfsprogs`, das unter Umständen extra installiert werden muss.

► `-A` bzw. `--alloc-start n`

lässt am Beginn des Devices `n` Byte ungenutzt (standardmäßig gilt `n=0`).

► `-d` bzw. `--data type`

gibt den gewünschten RAID-Typ für die Daten an. Erlaubte Werte sind momentan `raid0`, `raid1`, `raid10`, `raid5`, `raid6` oder `single` (gilt standardmäßig). Wenn Sie einen btrfs-RAID-Verbund einrichten, müssen Sie entsprechend viele Devices angeben (zwei für RAID-1, vier für RAID-10).

► `-l` bzw. `--label name`

gibt dem Dateisystem einen Namen.

► `-m` bzw. `--data type`

gibt den gewünschten RAID-Typ für die Metadaten an. Neben den bei der Option `-d` angegebenen RAID-Varianten gibt es zusätzlich `dup`. Diese Variante gilt standardmäßig bei Dateisystemen, die nur ein Device umfassen. Alle Metadaten werden daher aus Sicherheitsgründen doppelt gespeichert. Eine Ausnahme sind SSDs, wo bei Ein-Device-Dateisystemen per Default `single` gilt. `dup` ist hier nicht zweckmäßig, weil die interne Optimierung vieler SSDs die Redundanz der Daten erkennt und diese nur einfach abspeichert. Damit geht der durch die Duplizierung beabsichtigte Sicherheitsgewinn wieder verloren.

Wenn Sie ein `btrfs`-Dateisystem ohne weitere Angabe von Optionen über mehrere Devices verteilen, gilt standardmäßig der RAID-Typ `raid1`. Somit werden die Daten einfach gespeichert, die Metadaten aber mehrfach!

Beim Erstellen eines RAID-Verbunds geben Sie in der Regel mit `-m` den gleichen RAID-Typ wie bei der Option `-d` an. Wenn Ihr Dateisystem so schnell wie möglich sein soll, verwenden Sie `-d raid0 -m raid0`, um alle Redundanzen zu eliminieren.

### Beispiel

Ein RAID-1-Dateisystem, das die beiden möglichst gleich großen Partitionen `/dev/sdb1` und `/dev/sdc1` nutzt, richten Sie mit diesem Kommando ein:

```
root# mkfs.btrfs -d raid1 -m raid1 /dev/sdb1 /dev/sdc1
```

Sollte ein Device ausfallen, können Sie das Dateisystem mit der zusätzlichen `mount`-Option `degraded` weiterverwenden:

```
root# mount -o degraded /dev/sdb1 /media/btrfs
```

Um den RAID-Verbund wiederherzustellen, fügen Sie dem Dateisystem wieder ein neues Device hinzu. Im folgenden Beispiel ist das wieder `/dev/sdc1`, wobei diese Partition nun aber von einer neuen Festplatte oder SSD stammt. Um das Dateisystem wieder über beide Devices zu verteilen und somit die RAID-1-Redundanz wiederherzustellen, müssen Sie außerdem `btrfs filesystem balance` ausführen. Bei großen Dateisystemen dauert die Ausführung dieses Kommandos naturgemäß sehr lange. Immerhin kann das Dateisystem in dieser Zeit genutzt werden, wenn auch mit stark verminderter Geschwindigkeit.

```
root# btrfs device add /dev/sdc1 /media/btrfs
root# btrfs filesystem balance /media/btrfs
```

Erst jetzt kann das defekte Device aus dem Dateisystem entfernt werden. Dabei verwenden Sie zur Device-Angabe das Schlüsselwort `missing`:

```
root# btrfs device delete missing /media/btrfs
```

```
mkfs.exfat [optionen] device
```

`mkfs.exfat` aus dem Paket `exfat-utils` richtet ein exFAT-Dateisystem ein. Dieses Dateisystem ist speziell für große SD-Karten optimiert. Es kommt standardmäßig bei SDXC-Speicherkarten zum Einsatz und wird von SDXC-kompatiblen Digitalkameras und Smartphones unterstützt.



► **-i id**

bestimmt eine 32-Bit-ID-Nummer für das Dateisystem. Die Zahl wird als hexadezimaler Code angegeben. Bei der Ausführung des Kommandos ohne die Option wird eine zufällige ID festgelegt.

► **-n name**

legt den Namen (Label) für das Dateisystem fest. Standardmäßig bleibt der Name leer. Der Name kann bei Bedarf nachträglich mit exfatlabel verändert werden.

```
mkfs.ext2 [optionen] device [blocks]
mkfs.ext3 [optionen] device [blocks]
mkfs.ext4 [optionen] device [blocks]
mke2fs [optionen] device [blocks]
```

mke2fs bzw. mkfs.ext2, mkfs.ext3 oder mkfs.ext4 richten ein ext2-, ext3 oder ext4-Dateisystem ein. Alle vier Kommandos verweisen durch Links auf dasselbe Programm, verwenden aber unterschiedliche Standardoptionen, die in `/etc/mk2efs.conf` gespeichert sind.

► **-b n**

bestimmt die Blockgröße (normalerweise 4096 Bytes). Für *n* sind die Werte 1024, 2048, 4096 etc. erlaubt (also  $2^n \times 1024$ ).

► **-c**

testet vor dem Einrichten des Datenträgers mit `badblocks`, ob defekte Blöcke existieren. Derartige Blöcke werden für das Dateisystem nicht genutzt. Der Test erfordert, dass *jeder* Datenblock verändert und wieder gelesen werden muss, und verlängert daher die Zeit zum Einrichten des Dateisystems erheblich.

► **-i n**

gibt an, nach jeweils wie vielen Bytes ein I-Node eingerichtet wird. (I-Nodes sind interne Verwaltungseinheiten eines Dateisystems. In einem I-Node werden alle Verwaltungsinformationen einer Datei mit Ausnahme des Namens – also Zugriffsbits, Besitzer, Datum des letzten Zugriffs etc. – gespeichert. Die Anzahl der I-Nodes wird beim Formatieren unveränderlich festgelegt.) Beispielsweise gibt es bei  $n=4096$  pro MByte  $1.048.576 / 4096 = 256$  I-Nodes, d. h., es können pro MByte maximal 256 Dateien oder Verzeichnisse gespeichert werden, egal wie klein die Dateien sind.

Wenn Sie sehr viele sehr kleine Dateien speichern möchten, sollten Sie  $n$  kleiner wählen. Der Minimalwert beträgt 1024 (das entspricht 2048 Dateien pro MByte). Es ist nicht sinnvoll, den Wert kleiner als die Blockgröße zu wählen, da pro Block ohnedies nur eine Datei gespeichert werden kann. Wenn Sie dagegen nur sehr wenige sehr große Dateien speichern möchten, können Sie  $n$  auch größer wählen. Sie verringern damit den Verwaltungs-Overhead.

► -J

ermöglicht die Übergabe zusätzlicher Parameter, etwa um die Größe der Journaling-Datei einzustellen oder um die Journaling-Datei in einem anderen Device einzurichten. Das kann unter Umständen die Geschwindigkeit des Dateisystems ein wenig verbessern, macht aber gleichzeitig die Administration und die Wiederherstellung des Dateisystems nach einem Crash komplizierter. Die zur Auswahl stehenden Optionen werden in der Manual-Seite von `mke2fs` ausführlich beschrieben.

► -m  $n$

gibt an, wie viel Prozent des Datenträgers für Daten von `root` reserviert werden sollen (standardmäßig 5 Prozent). Dieser Reservespeicher ermöglicht es `root`, selbst dann noch zu arbeiten, wenn das Dateisystem für alle anderen Benutzer bereits restlos voll ist. Es handelt sich also um eine Sicherheitsreserve.

► -t ext2/ext3/ext4

gibt die gewünschte ext-Version an (nur bei `mke2fs` erforderlich).

## Beispiel

Die beiden folgenden Kommandos richten zuerst ein neues ext4-Dateisystem in der Partition `/dev/sda5` ein und deaktivieren dann die automatische Überprüfung des Dateisystems (siehe auch die Beschreibung des Kommandos `tune2fs`):

```
root# mkfs.ext4 /dev/sda5
root# tune2fs -i 0 -c 0 /dev/sda5
```

**mkfs.ntfs** [optionen] device

`mkfs.ntfs` aus dem Paket `ntfsprogs` richtet ein NTFS-Dateisystem ein, das kompatibel mit allen aktuellen Windows-Versionen ist. Das Kommando ist bei vielen Distributionen Teil des Pakets `ntfsprogs` und muss extra installiert werden.

- `-f` bzw. `--fast`

verzichtet darauf, jeden Datenblock mit 0-Bytes vollzuschreiben. Die Option beschleunigt die Formatierung um ein Vielfaches und ist unbedingt empfehlenswert.

```
mkfs.vfat [optionen] device
```

`mkfs.vfat` richtet ein Windows-Dateisystem ein, das kompatibel mit allen gängigen Windows-Versionen ist. Dieses Format bietet sich vor allem für USB-Sticks und SD-Karten an, die auch unter Windows oder in Kameras verwendet werden.

- `-F 12/16/32`

gibt den Typ der FAT-Tabellen an (12, 16 oder 32 Bit). Normalerweise wählt `mkfs.vfat` bei großen Datenträgern automatisch den 32-Bit-Modus. Wenn nicht, hilft diese Option.

- `-I`

erlaubt die Formatierung des gesamten Datenträgers, ohne dass eine Partitionstabelle eingerichtet wird (`mkfs.vfat -I /dev/sdg`). Das ist mitunter bei USB-Sticks üblich (»Superfloppy«-Format).

- `-n name`

gibt dem Dateisystem einen maximal 11 Zeichen langen Namen (Label).

## Beispiel

Das folgende Kommando richtet in der Partition `/dev/sdd1` ein 32-Bit-VFAT-Dateisystem mit dem Namen `MeineFotos` ein:

```
root# mkfs.vfat -F 32 -n MeineFotos /dev/sdd1
```

```
mkfs.xfs [optionen] device
```

`mkfs.xfs` richtet ein xfs-Dateisystem ein. Dieser Prozess kann durch unzählige Optionen gesteuert werden (`man mkfs.xfs`), von denen hier die wichtigsten herausgegriffen sind:

- `-b size=n`

gibt die gewünschte Blockgröße an (standardmäßig 4096 Byte).

- ▶ `-l logdev=device`  
speichert die Journaling-Daten in einer eigenen Partition.
- ▶ `-l size=n`  
gibt die Größe des Bereichs für die Journaling-Daten an. Ohne diese Option wählt `mkfs.xfs` selbst eine geeignete Größe.
- ▶ `-L label`  
gibt dem Dateisystem den gewünschten Namen (maximal 12 Zeichen).

### mkinitrd

`mkinitrd` war bei alten Distributionen für das Erzeugen einer `initrd`-Datei zuständig. Aktuelle Distributionen verwenden stattdessen `dracut` (CentOS, Fedora, RHEL, SUSE) oder `update-initramfs` (Debian, Ubuntu).

### `mknod` `devicedatei {bc}` `major minor`

`mknod` richtet eine neue Device-Datei ein. Device-Dateien befinden sich im Verzeichnis `/dev` und ermöglichen den Zugriff auf diverse Hardware-Komponenten. Device-Dateien werden durch drei Informationen charakterisiert: `major` und `minor` geben durch zwei Zahlen den Treiber an, mit dessen Hilfe auf die Geräte zugegriffen werden kann (Major- und Minor-Device-Nummer). Die Zeichen `b` oder `c` geben an, ob das Gerät gepuffert oder ungepuffert arbeitet.

Bei allen aktuellen Linux-Distributionen werden Device-Dateien automatisch durch das `udev`-System eingerichtet. Eine manuelle Ausführung von `mknod` ist daher nicht erforderlich. Eine Liste der wichtigsten Linux-Devices und der dazugehörigen Device-Nummern finden Sie hier:

<http://www.kernel.org/doc/Documentation/devices.txt>

### `mkpasswd` [`optionen`]

`mkpasswd` aus dem Paket `expect` (CentOS, Fedora, RHEL) generiert zufällige Passwörter.

- ▶ `-c n`  
gibt an, wie viele Kleinbuchstaben mindestens im Passwort enthalten sein müssen (Default: 2).

- ▶ `-C n`  
gibt an, wie viele Großbuchstaben mindestens im Passwort enthalten sein müssen (Default: 2).
- ▶ `-d n`  
gibt an, wie viele Ziffern mindestens im Passwort enthalten sein müssen (Default: 2).
- ▶ `-l n`  
gibt die gewünschte Zeichenanzahl des Passworts an (standardmäßig 9).
- ▶ `-s n`  
gibt an, wie viele Sonderzeichen mindestens im Passwort enthalten sein müssen.

### Beispiel

Das folgende Kommando erzeugt ein Sechs-Zeichen-Passwort ohne Sonderzeichen:

```
root# mkpasswd -l 6 -s 0
SX39vz
```

**mkswap** device / datei

mkswap richtet ein Device (z.B. eine Festplattenpartition) oder eine Datei als Swap-Bereich ein. mkswap kann nur von root ausgeführt werden. Der Swap-Bereich kann anschließend mit swapon aktiviert werden. Damit der Swap-Bereich bei jedem Rechnerstart automatisch genutzt wird, muss er in die Datei `/etc/fstab` eingetragen werden.

### Beispiel

Die folgenden Kommandos füllen zuerst eine 512 MByte große Datei mit Nullen, machen die Datei zur Swap-Datei und aktivieren diese schließlich:

```
root# dd if=/dev/zero of=swapfile bs=1M count=512
root# chmod 600 swapfile
root# mkswap swapfile
root# swapon swapfile
```

```
mkudffs [optionen] device
```

mkudffs richtet auf einem optischen Datenträger ein UDF-Dateisystem ein. Der Datenträger muss vorher mit dvd+rw-format formatiert werden.

- ▶ `--media-type=medientyp`  
gibt den Medientyp an. Zulässige Werte sind unter anderem `hd` (Festplatte), `cdrw`, `dvd` (DVD-R[W], DVD+R[W]) und `dvdrw`.
- ▶ `--udfrev=version`  
gibt die gewünschte UDF-Revisionsnummer an. Zulässige Werte sind `0x0102`, `0x0150`, `0x0200` und `0x0201` (gilt standardmäßig).
- ▶ `--vid=name`  
gibt den Datenträgernamen an (Volume ID).

```
modinfo modulname
```

modinfo liefert Informationen über das angegebene Modul. Dazu zählen der vollständige Dateiname, der Autor, die Lizenz, abhängige Module, eine Kurzbeschreibung sowie eine Liste aller Parameter, die das Modul kennt.

```
modprobe [optionen] modulname [parameter=wert ...]
```

modprobe lädt das angegebene Modul in den Kernel. Das Kommando ist eine erweiterte Variante von insmod. Es beachtet die in `/usr/lib/modules/kernel-version/modules.dep` definierten Modulabhängigkeiten und lädt gegebenenfalls auch Module, die vom gewünschten Modul vorausgesetzt werden. Außerdem berücksichtigt es die in `/etc/modprobe.d/*` angegebenen Modulparameter.

- ▶ `-c`  
liefert eine schier endlose Liste aller zurzeit gültigen Modulooptionen und Moduleinstellungen. Die Parameter ergeben sich sowohl aus dem Quelltext des `modutils`-Pakets als auch aus den Einstellungen in `/etc/modprobe.d/*`.
- ▶ `-f`  
erzwingt das Laden des Moduls selbst dann, wenn es für eine andere Kernelversion kompiliert wurde. Ob das tatsächlich funktioniert, hängt davon ab, ob es zwischen der Kernel- und der Modulversion irgendwelche Inkompatibilitäten gibt.

► -r

entfernt das Modul aus dem Kernel (anstatt es zu laden).

**mogrify** [optionen] bilddatei

**mogrify** verändert Parameter einer Bilddatei, beispielsweise die Auflösung oder die Anzahl der Farben. Dabei kommen dieselben Optionen wie bei **convert** zum Einsatz. Im Gegensatz zu **convert**, das eine neue Datei erzeugt, überschreibt **mogrify** die ursprüngliche Datei. Das Kommando ist Teil des Image-Magick-Pakets.

### Beispiel

Das folgende Kommando reduziert die Auflösung eines Bilds auf 800 × 600 Pixel:

```
user$ mogrify -resize 800x600 foto.jpg
```

**more** datei

**more** zeigt den Inhalt einer Textdatei seitenweise an. Nach jeder Seite wird die Anzeige unterbrochen. **more** wartet jetzt auf eine Tastatureingabe. Die wichtigsten Eingabemöglichkeiten sind **↓** (eine Zeile nach unten), die Leertaste (eine Seite nach unten), **↑** (eine Seite nach oben) und **q** (Quit). Statt **more** wird in der Regel das leistungsfähigere Kommando **less** eingesetzt.

**mount**

**mount** [optionen] device verzeichnis

**mount** ohne Parameter liefert eine Liste aller momentan eingebundenen Dateisysteme inklusive des Dateisystemtyps und der **mount**-Optionen. In der zweiten Syntaxvariante bindet **mount** einen Datenträger (Partition einer Festplatte, USB-Memorystick, Daten-DVD ...) in das Linux-Dateisystem ein. Dabei müssen Sie die Device-Bezeichnung des Datenträgers (beispielsweise **/dev/sdc1**) und das Verzeichnis angeben, bei dem das Dateisystem in den Verzeichnisbaum eingebunden wird.

Das Einbinden von Dateisystemen beim Start von Linux wird durch die Datei **/etc/fstab** gesteuert. Für alle in **/etc/fstab** verzeichneten Devices kann **mount** in einer Kurzform verwendet werden, bei der nur die Device-Datei oder das Mount-Verzeichnis angegeben wird. **mount** liest die fehlenden Daten und Optionen selbstständig aus **fstab**.

Datenträger werden mit `umount` wieder aus dem Dateisystem entfernt (siehe [umount](#)). Sowohl `mount` als auch `umount` dürfen im Regelfall nur von `root` ausgeführt werden. Eine Ausnahme stellen Laufwerke dar, die in `fstab` mit der Option `user` oder `users` gekennzeichnet sind.

Die folgende Beschreibung allgemeiner und dateisystemspezifischer `mount`-Optionen erstreckt sich über mehrere Seiten. Im Anschluss daran folgen einige Beispiele.

### Allgemeine mount-Optionen

- ▶ `--move olddir newdir`  
ändert das Verzeichnis, an dem ein Dateisystem eingebunden ist.
- ▶ `-o optionen`  
ermöglicht die Angabe zusätzlicher Optionen, z. B. in der Form `-o acl,user_xattr`. Die so angegebenen Optionen dürfen nicht durch Leerzeichen getrennt werden! Eine Referenz der Dateisystemoptionen folgt unten.
- ▶ `-r`  
verhindert Schreiboperationen auf dem Datenträger (*read-only*).
- ▶ `-t dateisystem`  
gibt das Dateisystem an. Infrage kommen unter anderem `ext2/3/4` sowie `btrfs` oder `xfs` für Linux-Partitionen, `vfat` und `ntfs` für Windows-Partitionen sowie `iso9660` und `udf` für Daten-CDs und -DVDs.

### Allgemeine Dateisystemoptionen

Mit `-o` können Sie unzählige Optionen übergeben, die das Verhalten des eingebundenen Dateisystems beeinflussen. Im Folgenden werden die Schlüsselwörter für allgemeine Optionen zusammengefasst. Dateisystemspezifische Optionen folgen etwas weiter unten.

Alle Optionen für `mount -o` sind auch in der vierten Spalte von `/etc/fstab` zulässig. Diese Datei wird während des Bootprozesses ausgewertet, unter Ubuntu durch das Kommando `mountall`. Die folgende Zusammenstellung gilt daher auch für `/etc/fstab` und für `mountall` und wurde deshalb um einige `fstab`- bzw. `mountall`-spezifische Schlüsselwörter ergänzt, die für das Kommando `mount` nicht unmittelbar von Bedeutung sind. Dazu zählen `defaults`, `noauto`, `nobootwait`, `optional`, `owner`, `user` und `users`.

- ▶ `atime / noatime / relatime / strictatime`  
gibt an, unter welchen Umständen die *Inode Access Time* einer Datei aktualisiert wird: bei jedem Lesezugriff (`atime` bzw. `strictatime`, POSIX-konform), maximal



einmal pro Tag (*relatime*) oder nie (*noatime*). Bei Schreibzugriffen wird die *Inode Access Time* in jedem Fall geändert.

Die Option hat einen relativ großen Einfluss auf die Geschwindigkeit von Leseoperationen und auf die Lebensdauer von Flash-Datenspeichern. Standardmäßig gilt bis Kernel 2.6.29 *atime*, bei aktuelleren Kernelversionen aber die schnellere Einstellung *relatime*! Wenn Sie das E-Mail-Programm *mutt* einsetzen, dürfen Sie die Einstellung *noatime* nicht verwenden! *relatime* verursacht hingegen keine Probleme.

► *defaults*

gibt an, dass das Dateisystem mit den Standardoptionen in den Verzeichnisbaum eingebunden werden soll. *defaults* wird in der vierten Spalte von */etc/fstab* angegeben, wenn keine anderen Optionen verwendet werden sollen. (Die Spalte darf nicht leer bleiben.)

► *dev / nodev*

bewirkt bzw. verhindert, dass Device-Dateien als solche interpretiert werden. Unter Linux können Dateien als Block- oder Character-Devices gekennzeichnet werden. Die Option ist aus Sicherheitsgründen für CD/DVD-Laufwerke sowie für externe Datenträger sinnvoll.

► *discard*

aktiviert bei den Dateisystemen *ext4*, *btrfs* und *xfs* die Trim-Funktion. Der Kernel informiert nun die Solid State Disk, welche Datenblöcke gelöscht wurden, und gibt der SSD so die Möglichkeit, die interne Verwaltung der Speicherzellen zu optimieren. Die Trim-Funktion funktioniert auch in Kombination mit LVM und seit Kernel 3.7 in Kombination mit Software-RAID (*mdraid*).

Ob man die *discard*-Option einsetzen soll, ist umstritten: Sie hilft zwar bei der langfristigen Optimierung der SSD, kann aber zu einer starken Verlangsamung von Löschoperationen führen (etwa wenn in kurzer Zeit viele Dateien gelöscht werden sollen, siehe beispielsweise <https://patrick-nagel.net/blog/archives/337>). Eine gute Alternative zur *discard*-Option ist ein Cron-Job, der einmal täglich oder einmal wöchentlich das Kommando *fstrim* ausführt und auf diese Weise alle gelöschten Sektoren an die SSD meldet.

► *exec / noexec*

legt fest, ob auf dem Dateisystem befindliche Programme unter Linux ausgeführt werden dürfen oder nicht. Bei den meisten Dateisystemen gilt standardmäßig die Einstellung *exec*. Sicherheitsbewusste Administratoren werden für CD/DVD-Laufwerke und externe Datenträger aber die Option *noexec* angeben. (Wenn Sie die

Option `user` oder `users` verwenden, gilt `noexec`. Das kann durch eine `exec`-Option wieder geändert werden.)

► `noauto`

ist nur in `/etc/fstab` zweckmäßig und bewirkt dort, dass der in dieser Zeile genannte Datenträger beim Systemstart *nicht* automatisch eingebunden wird. Dennoch ist es sinnvoll, den Datenträger in `fstab` einzutragen, weil die Anwender nun bequem `mount name` durchführen können, ohne dabei alle weiteren `mount`-Optionen explizit anzugeben. `noauto` kommt beispielsweise für selten benötigte Datenpartitionen zum Einsatz.

► `nobootwait`

hält den Bootprozess nicht auf, wenn das Device nicht zur Verfügung steht. Das Schlüsselwort wird unter Ubuntu von `mountall` ausgewertet und kann in anderen Distributionen nicht verwendet werden. Es bietet sich vor allem für externe Datenträger an, die nicht immer angeschlossen sind, z. B. für eine USB-Festplatte.

► `nofail`

meldet keinen Fehler, wenn das Device gar nicht existiert. Die Option ist praktisch, wenn `mount` in einem Script versuchen soll, auf einen USB-Stick oder ein Netzwerkverzeichnis zuzugreifen.

► `optional`

ignoriert den `fstab`-Eintrag, wenn der Dateisystemtyp während des Bootprozesses nicht zur Verfügung steht. Das Schlüsselwort wird unter Ubuntu von `mountall` ausgewertet und kann in anderen Distributionen nicht verwendet werden.

► `owner`

erlaubt es jedem Benutzer, das betreffende Dateisystem selbst einzubinden bzw. wieder zu lösen. Die Option ist nur in `/etc/fstab` zweckmäßig. Der Unterschied zu `user` besteht darin, dass `mount` bzw. `umount` nur dann ausgeführt werden darf, wenn der Benutzer Zugriffsrechte auf die betreffende Device-Datei hat (z. B. `/dev/fd0`).

► `remount`

ändert die Optionen eines bereits in den Verzeichnisbaum eingebundenen Dateisystems. Diese Option kann nur bei einem direkten Aufruf von `mount` angegeben werden und ist in `/etc/fstab` nicht zulässig.

► `ro` / `rw`

gibt an, dass Dateien nur gelesen bzw. auch verändert werden dürfen. Normalerweise gilt für die meisten Dateisystemtypen `rw` (also *read-write*).

► `suid / nosuid`

erlaubt bzw. verhindert die Auswertung des `sid`- und `gid`-Zugriffsbits. Diese Zugriffsbits erlauben es gewöhnlichen Benutzern, Programme mit `root`-Rechten auszuführen. Dies ist häufig ein Sicherheitsrisiko. Die Option `nosuid` verhindert die Auswertung derartiger Zugriffsbits und sollte für CDs, DVDs und externe Datenträger genutzt werden.

► `sync`

bewirkt, dass Änderungen sofort gespeichert werden, anstatt sie für einige Sekunden im RAM zwischenspeichern und erst später auf den Datenträger zu übertragen. `sync` minimiert die Gefahr des Datenverlusts, wenn Sie einen Datenträger (USB-Stick) irrtümlich ohne `umount` entnehmen bzw. die Kabelverbindung lösen. `sync` hat allerdings den Nachteil, dass das Schreiben von Daten je nach Datenträger sehr viel ineffizienter erfolgt. Insbesondere bei USB-Memorysticks und -Festplatten sinkt die Geschwindigkeit um den Faktor zehn und mehr!

► `user`

ermöglicht es normalen Benutzern, das betreffende Dateisystem mit `umount` ab- und mit `mount` wieder anzumelden. Dazu ist ansonsten nur `root` in der Lage. Datenträger dürfen nur von dem Benutzer wieder aus dem Dateisystem gelöst werden, der `mount` veranlasst hat. Die Option ist nur in `/etc/fstab` zweckmäßig und vor allem für externe Datenträger gedacht. Wenn Sie `user` verwenden, gelten automatisch auch die Optionen `noexec`, `nosuid` und `nodelv`.

► `users`

hat dieselbe Bedeutung wie `user`, aber mit einem kleinen Unterschied: Jeder Benutzer darf mit `users` gekennzeichnete Datenträger wieder aus dem Dateisystem entfernen (`umount`). `users` erlaubt also, dass der Benutzer A `mount` ausführt und der Benutzer B später `umount` aufruft.

## btrfs-Optionen

Viele `mount`-Optionen hängen vom jeweiligen Dateisystem ab. Im Folgenden sind daher die wichtigsten dateissystemspezifischen Optionen zusammengefasst. Wir beginnen mit `btrfs`.

► `compress=zlib|lzo`

komprimiert alle neuen bzw. veränderten Dateien. Vorhandene Dateien bleiben unverändert, solange sie nur gelesen werden. Was die Kompressionsmethode betrifft, haben Sie die Wahl zwischen dem etwas langsameren, aber besser komprimierenden `zlib`-Algorithmus und dem schnelleren, aber weniger platzsparenden

lzo-Algorithmus. In Zukunft wird `btrfs` voraussichtlich weitere Kompressionsverfahren unterstützen. Beinahe fertig ist die Integration des `snappy`-Algorithmus, der bei ähnlichen Kompressionsraten etwas schneller als das `lzo`-Verfahren sein soll.

► `compress-force=zlib|lzo`

komprimiert Dateien selbst dann, wenn dies nicht erfolgversprechend ist. Normalerweise erkennt das `btrfs`-Dateisystem bereits komprimierte Dateien und verzichtet darauf, diese nochmals zu komprimieren, weil dies in der Regel keine Platzersparnis mit sich bringt, aber eine Menge CPU-Zeit kostet. Die Verwendung von `compress-force` statt `compress` deaktiviert diese Erkennung.

► `degraded`

bindet das Dateisystem selbst dann ein, wenn ein Teil der RAID-Devices fehlt. Das funktioniert nur, wenn aufgrund von RAID-Redundanzen alle Daten zur Verfügung stehen, also ein degradierter RAID-Verbund vorliegt.

► `noacl`

deaktiviert die ACL-Zugriffsverwaltung.

► `nobarrier`

deaktiviert die sogenannten *Write Barriers*. Eine Write Barrier blockiert weitere Schreibvorgänge, bis alle bisherigen Schreibvorgänge physikalisch abgeschlossen sind. Ohne Barriers ist das Dateisystem schneller, gleichzeitig steigt aber das Risiko eines Datenverlusts bei einem Crash, insbesondere bei Dateisystemen, die über mehrere Devices verteilt sind (RAID). Barriers funktionieren nur, wenn die Festplatte die Schreibaufforderung auch tatsächlich befolgt – was (aus Performance-Gründen) häufig nicht der Fall ist.

► `ssd`

optimiert Dateioperationen für Solid State Disks (SSDs). Die Option ist automatisch aktiv, wenn der `btrfs`-Treiber feststellen kann, dass es sich beim Datenträger um eine SSD handelt. Die Option hat allerdings keinen Einfluss auf das Trim-Verhalten, das separat durch die Option `discard` gesteuert wird.

► `subvolume=name` oder `subvolid=n`

verwendet das angegebene Subvolume (und nicht das Default-Subvolume). Die Volume-ID für `subvolid` können Sie mit `btrfs subvolume list` ermitteln.

## CIFS-Optionen

Mit dem *Common Internet File System* (CIFS) können Sie Windows-Netzwerkverzeichnisse in das Dateisystem integrieren. CIFS löst SMBFS ab. Damit Sie CIFS nutzen können, müssen Sie bei manchen Distributionen ein eigenes Paket installieren, z. B. `cifs-utils` oder `smbfs`.

Einige Worte zu den Zugriffsrechten: Wenn die Dateien von einem gewöhnlichen Windows-PC stammen, werden sie unter Linux dem Benutzer `root` zugeordnet. Alle Benutzer dürfen die Dateien lesen, aber nur `root` darf sie verändern. Wenn ein gewöhnlicher Benutzer Zugriff auf die Dateien bekommen soll, erreichen Sie das durch die korrekte Einstellung der Optionen `uid`, `gid`, `file_mode` und `dir_mode`.

Wenn die Dateien hingegen von einem Datei-Server stammen, der die CIFS-Unix-Erweiterung unterstützt (das ist der Fall, wenn Samba auf einem Linux-Rechner läuft), dann werden die UIDs, GIDs und Zugriffsbits der Dateien unverändert weitergeleitet. Die Optionen `uid`, `gid`, `file_mode` und `dir_mode` bleiben dann wirkungslos, es sei denn, Sie deaktivieren die Unix-Erweiterungen mit der Option `nounix`.

Weitere Details zu den CIFS-spezifischen `mount`-Optionen gibt man `mount.cifs`.

### ► `credentials=datei`

gibt eine Datei an, die den Benutzernamen und das Passwort für den Login beim Windows- bzw. Samba-Server enthält. Damit vermeiden Sie die Angabe von Passwörtern in `/etc/fstab`. Die Datei enthält drei Zeilen mit `username=xxx` (nicht `user=xx!`), `password=xxx` und bei Bedarf `domain=xxx`.

### ► `dir_mode=n` (`dmask` bei SMBFS)

gibt an, welche Zugriffsrechte für den Zugriff auf Verzeichnisse gelten. `dir_mode=0770` gibt dem durch `uid` angegebenen Benutzer sowie allen Mitgliedern der Gruppe `gid` uneingeschränkte Rechte (Verzeichnis lesen, öffnen und ändern).

### ► `domain=workgroup`

gibt die Domäne oder Arbeitsgruppe an.

### ► `file_mode=n` (`fmask` bei SMBFS)

gibt an, welche Zugriffsrechte für den Dateizugriff gelten. `n` muss mit einer Null beginnen und enthält die Rechte in der von `chmod` vertrauten Oktalschreibweise. `file_mode=0444` gibt allen Benutzern Lese- und Schreibrechte. Die Einstellung wird ignoriert, wenn der Datei-Server die CIFS-Unix-Erweiterung unterstützt.

► `iocharset=name`

gibt den gewünschten Zeichensatz an. Wenn deutsche Sonderzeichen oder andere Nicht-ASCII-Zeichen in Dateinamen falsch angezeigt werden, hilft in der Regel `iocharset=utf8`.

► `nodfs`

deaktiviert die DFS-Funktionen. DFS steht für *Distributed File System* und erlaubt es, auf die Netzwerkverzeichnisse unterschiedlicher Server einheitlich zuzugreifen. Einige alte NAS-Festplatten verwenden ebenso alte Samba-Versionen mit bekannten DFS-Fehlern, die insbesondere beim Einsatz des Kommandos `rsync` zu Problemen führen. Abhilfe schafft in solchen Fällen die `nodfs`-Option.

► `nounix`

deaktiviert den Unix-Kompatibilitätsmodus. Zum Verständnis dieser Option sind ein paar Hintergrundinformationen unumgänglich. Der CIFS-Standard wurde nämlich um die sogenannten Unix Extensions erweitert, um eine bessere Kompatibilität mit Unix/Linux-Systemen zu erzielen. Bei manchen NAS-Geräten sind diese Extensions auch aktiv. In diesem Fall übernimmt `mount` die Benutzer- und Gruppeninformationen sowie die Zugriffsrechte vom NAS-Gerät und ignoriert die Optionen `uid`, `gid`, `dir_mode` und `file_mode`.

Das klingt auf den ersten Blick praktisch, ist es aber zumeist nicht: Die Unix Extensions ergeben nur dann einen Sinn, wenn die UIDs und GIDs des lokalen Rechners und des Samba-Servers abgestimmt sind. Das ist selten der Fall, bei NAS-Geräten praktisch nie. Daher muss der Kompatibilitätsmodus deaktiviert werden, damit `uid`, `gid`, `dir_mode` und `file_mode` wirksam bleiben. Mitunter können Sie mit `nounix` auch Inkompatibilitäten aus dem Weg gehen, die sich in der vagen Fehlermeldung *operation not supported* äußern.

► `password=pw`

gibt das Passwort für die Authentifizierung beim Windows- bzw. Samba-Server an. Die direkte Passwortangabe ist unsicher! Besser ist es, das Passwort in einer Datei anzugeben und auf diese mit dem Schlüsselwort `credentials` zu verweisen.

► `sec=modus`

gibt den Authentifizierungsmodus an. Dieser Modus steuert die Authentifizierung beim Samba-Server. Die Defaulteinstellung lautet `ntlmssp`. Dieses Schlüsselwort bezeichnet ein Windows-typisches Verfahren mit Passwort-Hashing, wobei die Daten in eine NTLMSSP-Nachricht verpackt werden. Alternative Einstellungen wie `ntlmssp` oder `krb5i` (Kerberos) beschreibt man `mount.cifs`.

► `uid=u,gid=g`

geben wie beim `vfat`-Treiber den Besitzer und die Gruppe der Dateien an. An die Optionen können statt der UID/GID-Nummern auch die Benutzer- bzw. Gruppennamen übergeben werden.

► `user=name` (username bei SMBFS)

gibt den Benutzernamen für die Authentifizierung beim Windows- bzw. Samba-Server an. An die Option können gleichzeitig auch der Arbeitsgruppenname und das Passwort übergeben werden (`user=workgroup/name%password`).

### exFAT-Optionen

Das exFAT-Dateisystem (`-t exfat`) unterstützt die Optionen `dmask`, `fmask`, `gid`, `noatime`, `ro` (Read-only) `uid` und `umask`. Die Optionen haben dieselbe Wirkung wie die gleichnamigen VFAT-Optionen, die einige Seiten weiter beschrieben werden. Einzig `noatime` steht bei VFAT-Dateisystemen nicht zur Verfügung. Diese Option verhindert, dass bei Lesezugriffen die Access-Time-Information aktualisiert wird, und verringert so die Anzahl der Schreiboperationen.

### ext3/ext4-Optionen

Die folgenden Optionen gelten für das Linux-Standarddateisystem `ext3` bzw. `ext4`:

► `acl`

aktiviert die ACL-Zugriffsverwaltung. Das erlaubt die Speicherung von zusätzlichen Zugriffsinformationen in Form von *Access Control Lists*. Die Option ist insbesondere beim Einsatz von Samba in heterogenen Netzwerken sinnvoll.

► `barrier=0` (nur für `ext4`)

deaktiviert die sogenannten *Write Barriers*, die sicherstellen, dass Änderungen in der richtigen Reihenfolge auf die Festplatte geschrieben werden. Das erhöht die Geschwindigkeit des Dateisystems, ist aber nur sicher, wenn der Festplatten-Cache durch eine Batterie gegen Stromausfälle und Abstürze abgesichert ist.

► `commit=n`

synchronisiert das Journal alle *n* Sekunden (standardmäßig alle 5 Sekunden). Das Paket `laptop-mode` vergrößert diese Zeitspanne bei Notebooks im Batteriebetrieb, um Energie zu sparen. Bei `ext3` mit `data=ordered` werden gleichzeitig mit dem Journal auch alle Dateiänderungen synchronisiert. Bei `ext3` mit `data=writeback` und bei `ext4` mit *Delayed Allocation* gilt das `commit`-Intervall hingegen nur für das Journal.

► `data=journal/ordered/writeback`

bestimmt den Journaling-Modus (Details dazu folgen gleich). Standardmäßig gilt für ext4 der Modus `ordered`. Für ext3 gilt standardmäßig für die Kernelversionen 2.6.30 bis 2.6.35 der Modus `writeback`, für alle Versionen davor und danach der sichere, aber etwas langsamere Modus `ordered`.

► `errors=continue/remount-ro/panic`

bestimmt das Verhalten, wenn bei der Überprüfung des Dateisystems ein Fehler auftritt. Das Defaultverhalten kann mithilfe des Befehls `tune2fs` mit der Option `-e` eingestellt werden.

► `grpuid`

bewirkt, dass neue Dateien automatisch die Gruppen-ID des Verzeichnisses übernehmen, in dem sie erzeugt wurden. Damit verhält sich das Dateisystem wie unter BSD bzw. so, als wäre in allen Verzeichnissen das SetGID-Bit gesetzt. (Standardmäßig ordnet Linux neuen Dateien die Gruppen-ID des Benutzers zu, der die Datei erzeugt.)

► `nodelalloc` (nur für ext4)

deaktiviert die *delayed allocation*.

► `noload`

ignoriert die vorhandene Journaling-Datei beim `mount`-Kommando. Das kann zweckmäßig sein, wenn die Journaling-Datei defekt ist.

► `sb=n`

verwendet Block *n* als Superblock (statt standardmäßig Block 1). Damit gelingt es in manchen Fällen, ein beschädigtes Dateisystem noch zu lesen. Normalerweise erstellt `mke2fs` alle 8192 Blöcke eine Kopie des Superblocks. Daher empfiehlt es sich, für *n* die Werte 8193, 16.385 etc. einzusetzen.

► `user_xattr`

aktiviert die Unterstützung für erweiterte Dateiattribute.

### ext3/ext4-Journaling-Modi

Das ext-Dateisystem kennt drei Verfahren, wie das Journaling durchgeführt wird:

► `data=ordered`

Bei diesem Modus werden im Journal nur Metadaten gespeichert, also Informationen *über* Dateien, keine Inhalte. Im Journal werden Dateien erst dann als *committed* gekennzeichnet, wenn sie vollständig gespeichert wurden. Nach einem Crash



kann das Dateisystem rasch wieder in einen konsistenten Zustand gebracht werden, weil unvollständig gespeicherte Dateien anhand des Journals sofort erkannt werden. Es ist aber unmöglich, solche Dateien wiederherzustellen.

Im Modus `data=ordered` wird das Journal alle fünf Sekunden mit der Festplatte synchronisiert. Bei `ext3` hat das zur Folge, dass sämtliche Änderungen an irgendwelchen Dateien innerhalb von fünf Sekunden physikalisch auf der Festplatte gespeichert werden. Dieses Standardverhalten ist zwar nicht besonders effizient, dafür aber sehr sicher: Selbst bei Totalabstürzen und Stromausfällen sind massive Datenverluste äußerst selten. `data=ordered` hat bei `ext3` aber eine unerfreuliche Nebenwirkung: Bei jedem Aufruf der `fsync`-Funktion wird nicht nur eine bestimmte Datei, sondern das gesamte Dateisystem synchronisiert. Das kann zu spürbaren Verzögerungen führen.

Bei `ext4` wird das Journal zwar ebenfalls alle fünf Sekunden synchronisiert, die eigentlichen Datenänderungen werden aber aufgrund der *Delayed Allocation* oft erst viel später gespeichert. Nur ein expliziter Aufruf der `fsync`-Funktion stellt die sofortige physikalische Speicherung einer Datei sicher! Glücklicherweise erfordert `fsync` bei `ext4` nicht, dass das gesamte Dateisystem synchronisiert werden muss. Die Funktion wird daher wesentlich schneller ausgeführt.

► `data=writeback`

Dieser Modus ähnelt dem `ordered`-Modus. Der einzige Unterschied besteht darin, dass das Journal und die Dateioperationen nicht immer vollständig synchron sind. Das Dateisystem wartet mit den *committed*-Einträgen im Journal nicht auf den Abschluss der Speicheroperation auf der Festplatte. Damit ist das Dateisystem etwas schneller als im `ordered`-Modus. Nach einem Crash ist die Integrität des Dateisystems weiterhin sichergestellt. Allerdings kann es vorkommen, dass veränderte Dateien alte Daten enthalten. Dieses Problem tritt nicht auf, wenn Anwendungsprogramme – wie im POSIX-Standard vorgesehen – den Speichervorgang mit `fsync` abschließen (siehe oben).

► `data=journal`

Anders als in den beiden anderen Modi werden jetzt im Journal auch die tatsächlichen Daten gespeichert. Dadurch müssen alle Änderungen *zweimal* gespeichert werden (zuerst in das Journal und dann in die betroffene Datei). Deswegen ist `ext3` in diesem Modus deutlich langsamer. Dafür können nach einem Crash Dateien wiederhergestellt werden, deren Änderungen bereits vollständig in das Journal (aber noch nicht in die Datei) eingetragen worden sind.

Grundsätzlich wird das Journal alle fünf Sekunden physikalisch auf der Festplatte gespeichert. Diese Zeitspanne kann durch die `mount`-Option `commit` verändert werden.

Intern kümmert sich der in den Kernel integrierte Journaling-Dämon `kjournald` um die regelmäßige Aktualisierung der Journaling-Datei.

### ISO-9660-Optionen (Daten-CDs/DVDs)

Daten-CDs und viele DVDs verwenden das Dateisystem ISO 9660 (`-t iso9660`).

- ▶ `iocharset=name`  
gibt den Zeichensatz an, mit dem Joliet-Dateinamen unter Linux verarbeitet werden sollen (standardmäßig ISO 8859-1). Die Option ist nur bei CDs sinnvoll, die die Joliet-Erweiterung nutzen.
- ▶ `norock, nojoliet`  
deaktiviert die Rock-Ridge- bzw. die Joliet-Erweiterung. (Der ISO-9660-Treiber erkennt und berücksichtigt diese Erweiterungen automatisch.)
- ▶ `session=n`  
gibt an, welche Session gelesen werden soll (nur bei Multi-Session-CDs).
- ▶ `uid=u,gid=g`  
geben den Besitzer und die Gruppe der Dateien an (standardmäßig jeweils `root`). Die Zugriffsbits können nicht eingestellt werden.
- ▶ `utf8`  
verwendet Linux-seitig den Zeichensatz UTF8, um Joliet-Dateinamen zu verarbeiten. Auch diese Option ist nur bei Joliet-CDs sinnvoll.

### NFS-Optionen

Das Dateisystem NFS (`-t nfs`) ist der übliche Weg zur Freigabe bzw. Nutzung von Netzwerkverzeichnissen unter Unix/Linux. Beachten Sie, dass sich `-t nfs` auf die NFS-Versionen 2 und 3 bezieht. Wenn Sie NFS4 nutzen möchten, müssen Sie im `mount`-Kommando `-t nfs4` angeben.

Die meisten NFS-Optionen gelten gleichermaßen für alle NFS-Versionen. Eine umfassende Zusammenfassung aller Optionen inklusive einer Differenzierung zwischen NFS2/3 und NFS4 gibt `man nfs`.

- ▶ `bg`  
bewirkt, dass der `mount`-Prozess im Hintergrund fortgesetzt wird, wenn der NFS-Server nicht erreichbar ist. Diese Option ist nur für `/etc/fstab` relevant.

- ▶ `hard`  
bewirkt, dass ein Programm, das auf eine NFS-Datei zugreift, hängen bleibt, wenn der NFS-Server nicht mehr zur Verfügung steht. (Die Alternative zu `hard` ist `soft`. Sie bewirkt, dass der Kernel es nach einer Weile aufgibt, die NFS-Datei wiederzufinden. Das klingt zwar sicherer, verursacht aber in der Realität noch mehr Probleme als `hard`.)
- ▶ `intr`  
ermöglicht es, ein Programm auch dann durch `kill` bzw. `[Strg]+[C]` zu stoppen, wenn eine offene NFS-Datei nicht mehr zur Verfügung steht. Die Option ist nur in Kombination mit `hard` gültig.
- ▶ `nfsvers=2/3`  
gibt die gewünschte NFS-Version an (nur für `-t nfs`).
- ▶ `rsiz=n,wsiz=n`  
gibt die Größe der Puffer für Lese- und Schreiboperationen an (in kByte). Der Standardwert lautet jeweils 4096.

Die richtigen `mount`-Optionen entscheiden maßgeblich über die mit NFS erzielbare Geschwindigkeit! Eine hohe Geschwindigkeit erzielen Sie mit den Optionen `hard`, `intr`, `rsiz=8192`, `wsiz=8192`. Wenn die NFS-Geschwindigkeit dennoch unter Ihren Erwartungen liegt, sollten Sie sicherstellen, dass `/etc/exports` auf dem NFS-Server die Option `async` verwendet.

## NTFS-Optionen

Die folgenden Optionen gelten für den `ntfs3g`-Treiber, der bei den meisten Distributionen für den Zugriff auf NTFS-Dateisysteme zum Einsatz kommt.

- ▶ `locale=name`  
gibt den lokalen Zeichensatz an (beispielsweise in der Form `locale=de_DE.UTF-8`). Die Option ist nur in `/etc/fstab` notwendig, wenn der Zeichensatz beim automatischen Einbinden der Dateisysteme durch den Init-V-Prozess noch nicht konfiguriert ist.
- ▶ `show_sys_files`  
zeigt auch NTFS-Systemdateien an.
- ▶ `streams_interface=none|windows|xattr`  
gibt an, wie der Zugriff auf Datei-Streams erfolgen soll: In der Standardeinstellung `xattr` erfolgt der Zugriff über die Kommandos `getfattr` bzw. `setfattr`.

windows erlaubt den Zugriff in der unter Windows üblichen Schreibweise `dateiname:streamname`. Die Standardeinstellung lautet `none`.

- ▶ `uid=u,gid=g,umask=m|fmask=f,dmask=d`

haben dieselbe Bedeutung wie beim `vfat`-Treiber (siehe unten). Als Dateibesitzer gilt der Benutzer, der `mount` ausgeführt hat. Alle Benutzer haben Lese- und Schreibzugriff!

### UDF-Optionen (DVDs)

Auf DVDs kommen zumeist die Dateisysteme ISO 9660 (siehe oben) oder UDF (`-t udf`) zum Einsatz. UDF ist auch das optimale Dateisystem für DVD-RAMs. UDF verwendet überwiegend dieselben Optionen wie ISO 9660. Im Folgenden sind deswegen nur zwei UDF-spezifische Optionen beschrieben:

- ▶ `undelete`  
gibt Zugriff auf gelöschte Dateien.
- ▶ `unhide`  
zeigt auch Dateien, die eigentlich versteckt sind.

### VFAT-Optionen

Mit `mount -t vfat` binden Sie Windows-VFAT-Partitionen in allen Varianten (FAT12, FAT16, FAT32) in den Verzeichnisbaum ein. Zahlreiche Optionen steuern, wer welche Zugriffsrechte auf die Dateien hat, wie die unterschiedlichen Zeichensätze in den Dateinamen verarbeitet werden etc.

- ▶ `codepage=name`  
gibt die Codeseite an (eine Art DOS-Zeichensatz), die VFAT-intern für kurze Dateinamen gilt (maximal 8+3 Zeichen). Lange Dateinamen werden auf jeden Fall im Unicode-Format gespeichert; für sie ist diese Option irrelevant. Aktuelle Windows-Versionen speichern auch kurze Dateinamen im VFAT-Format für lange Dateinamen, um so die korrekte Groß- und Kleinschreibung zu erhalten. Aus diesem Grund ist die Option zunehmend unwichtig. Der Standardwert der Option kann beim Kompilieren bestimmt werden und beträgt zumeist `cp437`.
- ▶ `flush`  
bewirkt, dass der Treiber sofort damit beginnt, zwischengespeicherte Daten auf dem Datenträger zu speichern. Die Option minimiert bei externen Datenträgern

(z. B. einem USB-Memorystick) die Zeit zum Speichern veränderter Daten. Im Gegensatz zur Option `sync` wartet das Dateisystem nicht auf den Abschluss der Synchronisation. Deswegen hat `flush` keinen negativen Einfluss auf die Geschwindigkeit von Schreiboperationen.

► `fmask=f,dmask=d`

stellt die Maske für Dateien (`fmask`) und Verzeichnisse (`dmask`) getrennt ein. Das ist vor allem deswegen praktisch, weil Sie oft für Verzeichnisse das x-Zugriffsbit setzen möchten, für Dateien aber nicht. Damit kann der Benutzer in alle Verzeichnisse wechseln, aber keine Dateien bzw. Programme ausführen.

Eine mögliche Einstellung ist `fmask=177,dmask=077`: Damit kann der Besitzer (`uid`) Dateien lesen und schreiben (`rw-----`) sowie in Verzeichnisse wechseln (`rw-x-----`). Alle anderen Benutzer haben keinerlei Zugriff auf die Dateien.

Ebenfalls oft zweckmäßig ist `fmask=133,dmask=022`: Damit gilt für Dateien `rw-r--r--` und für Verzeichnisse `rw-xr-x`, d. h., jeder darf alles lesen, aber nur der Besitzer darf etwas verändern.

► `iocharset`

gibt den Zeichensatz an, mit dem Windows-Dateinamen unter Linux verarbeitet werden sollen. Die Standardeinstellung lautet `iso8859-1`. Die Option kommt in der Regel nur dann zum Einsatz, wenn Sie unter Linux mit einem 8-Bit-Zeichensatz arbeiten. (VFAT-intern wird für lange Dateinamen in jedem Fall der Unicode-Zeichensatz verwendet. Die Option bestimmt nur, wie die Dateinamen unter Linux dargestellt werden.)

Wenn Sie unter Linux mit dem Zeichensatz UTF8 arbeiten, sollten Sie statt `iocharset` die im Folgenden beschriebene Option `utf8` verwenden! `iocharset=utf8` funktioniert zwar auf den ersten Blick ebenfalls, ist aber nicht empfehlenswert: Der VFAT-Treiber beachtet dann die exakte Groß- und Kleinschreibung, was den Windows-Gepflogenheiten widerspricht.

► `shortname=lower|win95|winnt|mixed`

gibt an, in welcher Groß- und Kleinschreibung kurze Dateinamen (maximal 8+3 Zeichen) präsentiert bzw. gespeichert werden. Das diesbezügliche Verhalten variierte unter Windows je nach Version.

Unter Linux gilt standardmäßig die Einstellung `lower`, d. h., kurze Dateinamen werden in Kleinbuchstaben präsentiert. Bei neuen Dateien wird der Name im VFAT-Format für lange Dateinamen gespeichert (auch wenn die DOS-Grenze von 8+3-Zeichen gar nicht überschritten wird), sodass die Groß- und Kleinschreibung erhalten bleibt. Ausgenommen von dieser Regel sind nur kurze Dateinamen,

die ausschließlich aus Großbuchstaben bestehen; derartige Namen werden DOS-typisch als kurze Dateinamen gespeichert.

Die Einstellung hat keinerlei Einfluss auf lange Dateinamen, deren Groß- und Kleinschreibung nach dem Erzeugen der Datei auf jeden Fall erhalten bleibt.

► `uid=u,gid=g,umask=m`

bestimmen, wem (`uid`) und welcher Gruppe (`gid`) die Windows-Dateien »gehören«, d. h., wer die Dateien lesen bzw. verändern darf. Üblicherweise geben Sie mit `uid` und `gid` die gewünschte Benutzer- bzw. Gruppennummer an. An `umask` übergeben Sie einen oktalen Zahlenwert mit der Bitmaske der *invertierten* Zugriffsbits. `umask=0` bedeutet daher, dass jeder alle Dateien lesen, schreiben und ausführen darf (`rxwxrwxrwx`). `umask=022` entspricht `rxwxr-xr-x`: Alle dürfen alle Dateien lesen, aber nur der Besitzer darf Dateien ändern.

Wenn diese Optionen nicht angegeben werden, gelten für das `vfat`-Dateisystem die folgenden Standardwerte:

- `uid`: UID des Benutzers, der `mount` ausführt. Normalerweise darf nur `root` das Kommando `mount` ausführen, es sei denn, die entsprechende Zeile in `/etc/fstab` enthält die Optionen `user`, `users` oder `owner`.
- `gid`: GID des Benutzers, der `mount` ausführt.
- `umask`: 022, d. h., alle dürfen alles lesen, aber nur der Besitzer darf Dateien bzw. Verzeichnisse verändern.

Oft enthält `fstab` die Einstellung `gid=users`. Damit gilt für `gid` die Standardgruppe des Benutzers, der `mount` ausführt. Ohne diese Option ist es möglich, zuerst mit `newgrp` eine andere Gruppe zu aktivieren, die dann auch für `mount` gilt.

► `utf8`

bewirkt, dass der Treiber die Dateinamen als UTF8-Zeichenketten an Linux zurückgibt. Die Option ist zur korrekten Darstellung von Nicht-ASCII-Zeichen erforderlich, wenn Sie unter Linux als Zeichensatz UTF8 verwenden.

## XFS-Optionen

Die meisten Optionen für das `xfs`-Dateisystem betreffen selten benutzte Funktionen (siehe `man mount`). ACLs und *Extended Attributes* stehen beim `xfs`-Dateisystem standardmäßig zur Verfügung und werden nicht durch Optionen gesteuert.

► `logdev=device`

gibt die Partition an, in der die Journaling-Daten gespeichert sind. Die Option ist nur notwendig, wenn beim Einrichten des Dateisystems eine externe Journaling-Partition angegeben wurde.

► `norecovery`

verhindert die Auswertung der Journaling-Daten beim Einbinden des Dateisystems. Gleichzeitig müssen Sie die Option `ro` verwenden (*Read-only*-Zugriff).

## Beispiele

Linux-Partitionen können Sie in der Regel ohne explizite Angabe des Dateisystems (also ohne die Option `-t`) in das Dateisystem integrieren:

```
root# mkdir /windows
root# mount /dev/sda7 /media/backup-partition
```

Das folgende Beispiel zeigt den Zugriff auf die Daten eines USB-Sticks über das Verzeichnis `/windows`:

```
root# mkdir /usbstick
root# mount -t vfat /dev/sdc1 /usbstick
```

Damit nicht nur `root`, sondern auch ein gewöhnlicher Benutzer mit der User-ID 1000 und der Gruppen-ID 1000 Zugriff auf den USB-Stick hat und Dateien schreiben oder verändern kann, sind mehr Optionen erforderlich:

```
root# mount -t vfat -o uid=1000,gid=1000,fmask=0022,dmask=0022 \
/dev/sdc1 /usbstick
```

Das nächste Kommando bindet das CD-ROM-Laufwerk mit einer Daten-CD (ISO-9660-Dateisystem) im Verzeichnis `/media/cdrom` in das Dateisystem ein:

```
root# mount -t iso9660 /dev/sr0 /media/cdrom
```

Das folgende Kommando bindet ein Netzwerkverzeichnis eines Windows- oder Samba-Servers, z. B. von einem NAS-Gerät, in den lokalen Verzeichnisbaum ein. Der Benutzer mit der UID/GID 1000 kann die Daten lesen und verändern.

```
root# mount -t cifs -o username=name,uid=1000,gid=1000,ioccharset=utf8,nounix \
//jupiter/myshare /media/winshare
Password for name: *****
```

Mit `mount -o remount` können Sie Optionen eines bereits eingebundenen Dateisystems verändern. Das folgende Kommando aktiviert beispielsweise die `exec`-Option für eine DVD, sodass darauf enthaltene Programme ausgeführt werden können:

```
root# mount /media/dvd -o remount,exec
```

Falls beim Einbinden der Systempartition während des Rechnerstarts Probleme auftreten, wird die Partition nur *read-only* eingebunden. Um die Fehlerursache – etwa einen falschen Eintrag in `/etc/fstab` – zu beheben, ist es aber oft erforderlich, Änderungen im Dateisystem durchzuführen. Dazu führen Sie das folgende Kommando

aus. Mit ihm wird die Systempartition neu eingebunden, wobei jetzt auch Schreibzugriffe möglich sind.

```
root# mount -o remount,rw /
```

**mpage** [optionen] quelldateien > zieldatei

mpage wandelt eine Textdatei in das PostScript-Format um. mpage kann auch vorhandene PostScript-Dateien platzsparend neu formatieren und anschließend beispielsweise mit zwei Seiten pro Blatt ausdrucken. Das funktioniert allerdings nicht mit allen PostScript-Dateien. mpage ist leider nicht Unicode-kompatibel.

► -1 -2 -4 -8

ordnet eine, zwei, vier oder acht Seiten auf einem Blatt an. Bei zwei oder acht wird das Querformat verwendet (normalerweise vier Seiten).

► -bA4

Ausgabe im DIN-A4-Format (normalerweise US-Letter).

► -C ISO-Latin.1

verwendet die ISO-Latin-1-Kodierung für den Zeichensatz. Die Option ist für den Ausdruck von Texten mit deutschen Sonderzeichen erforderlich. Andere Kodierungstabellen befinden sich in `/usr/lib/mpage`.

► -f

verteilt lange Zeilen auf mehrere Zeilen (statt sie abzuschneiden).

► -j *n-m*

druckt nur Blatt *n* bis *m* (normalerweise alle).

► -l

druckt im Querformat mit 55 Zeilen und 132 Spalten und einer entsprechend geringeren Schriftgröße (normalerweise Hochformat mit 66 Zeilen und 80 Spalten).

► -L *n*

druckt *n* Zeilen pro Seite und passt die Zeichensatzgröße entsprechend an (normalerweise 66 Zeilen).

► -o

verzichtet auf Rahmenlinien um die Seiten.



► -W *n*

druckt *n* Spalten pro Seite und passt die Zeichensatzgröße entsprechend an (normalerweise 80 Spalten).

**mt** [-f device] kommando

mt steuert Bandlaufwerke (Streamer). Als Standard-Device wird /dev/tape verwendet. Wenn Ihr Streamer nicht unter diesem Namen erreichbar ist, müssen Sie die Device-Datei exakt angeben, etwa durch -f /dev/nst0 für einen SCSI-Streamer. Die häufigste Anwendung von mt ist das Rückspulen von Bändern mit dem Kommando `rewind` bzw. das Auswerfen von Bändern mit `offline`. Mit `setblk` kann die Blockgröße verändert werden, und `stat` zeigt Statusinformationen zur Streamer-Einstellung an.

Es gibt zwei mt-Varianten: eine GNU-Variante und eine BSD-Variante. Mit den meisten Distributionen wird automatisch die BSD-Variante mitgeliefert. Der GNU-Variante fehlen viele Kommandos der BSD-Variante.

**mtr** [optionen] hostname

Das Kommando mtr sendet regelmäßig Netzwerkpakete zum angegebenen Host und analysiert die Antworten. Die Ergebnisliste kombiniert Daten von ping und traceroute. Beachten Sie, dass es zwei Versionen dieses Programms gibt: das hier beschriebene Textkommando und eine GTK-Variante mit grafischer Benutzeroberfläche. Bei Desktop-Installationen von Debian und Ubuntu ist standardmäßig die GTK-Variante installiert. Um stattdessen die Textversion zu installieren, führen Sie `apt-get install mtr-tiny` aus.

► -c *n*

führt *n* Tests aus. (Standardmäßig läuft das Programm, bis es mit `[Strg]+[C]` beendet wird.)

► -n bzw. --no-dns

verzichtet auf die Namensauflösung und gibt alle Zwischenstationen in Form von IP-Adressen an.

► -r bzw. --report

liefert nach Abschluss des Tests einen Bericht in Textform. Diese Option muss mit -c verbunden werden.

## Beispiel

Die folgenden Zeilen zeigen ein typisches `mtr`-Ergebnis, wobei die Verbindung zwischen einem lokalen ADSL-Anschluss und *google.de* getestet wurde:

```
user$ mtr -c 10 -r google.de
HOST: michael's-computer      Loss%  Snt  Last   Avg   Best  Wrst StDev
 1. |-- speedtouch.lan        0.0%   10  42.6  48.5   6.0  95.9  28.9
 2. |-- 178-191-207-254.adsl.high 0.0%   10  18.9  20.4  18.6  23.2   1.9
 3. |-- 195.3.74.129           0.0%   10  19.4  18.8  17.9  19.4   0.5
 4. |-- AUX10-GRAZBC10.highway.te 0.0%   10  21.2  21.3  20.7  22.0   0.3
 5. |-- 195.3.70.154          0.0%   10  21.2  27.3  20.9  81.2  18.9
...
 8. |-- 216.239.46.88          0.0%   10  25.8  26.3  25.8  27.7   0.6
 9. |-- bud01s08-in-f23.1e100.net 0.0%   10  25.7  25.8  25.0  26.8   0.5
```

```
multitail [optionen] datei1 [datei2 datei3 ...]
```

`multitail` hilft dabei, mehrere Logging-Dateien gleichzeitig zu überwachen. In der einfachsten Form führen Sie `multitail datei1 datei2 datei3 ...` aus. Damit wird das Terminal in mehrere gleich große Bereiche geteilt, die jeweils die letzten Zeilen der angegebenen Dateien zeigen. Wenn Sie den Dateien `-I` voranstellen, werden Änderungen mehrerer Dateien in einem Bereich gemeinsam ausgegeben. Das hat allerdings den Nachteil, dass schwer nachzuvollziehen ist, aus welcher Datei welche Änderungen stammen.

```
mv quelle ziel
mv dateien zielverzeichnis
```

`mv` benennt eine Datei oder ein Verzeichnis um bzw. verschiebt (eine oder mehrere) Dateien in ein anderes Verzeichnis.

- ▶ `-b` bzw. `--backup`  
benennt bereits vorhandene gleichnamige Dateien in Backup-Dateien (Name plus `~`-Zeichen) um, anstatt sie zu überschreiben.
- ▶ `-i` bzw. `--interactive`  
fragt, bevor vorhandene Dateien überschrieben werden.

## Beispiele

Das folgende Kommando verschiebt alle PDF-Dateien im aktuellen Verzeichnis in das Unterverzeichnis `pdf`:

```
user$ mv *.pdf pdf/
```

`mv` kann nicht dazu verwendet werden, mehrere Dateien umzubenennen. `mv *.xxx *.yyy` funktioniert also nicht. Um solche Operationen durchzuführen, müssen Sie `for` oder `sed` zu Hilfe nehmen. Entsprechende Beispiele finden Sie bei [cp](#).

```
mysql [optionen] [datenbankname] [ < name.sql]
```

Der Kommandointerpreter `mysql` führt SQL-Kommandos auf einem MySQL- oder MariaDB-Server aus, wahlweise interaktiv oder im Batch-Betrieb aus der Standardeingabe. Die Kommandos müssen mit einem Strichpunkt abgeschlossen werden. In der Batch-Variante wird `mysql` häufig zum Einlesen von Backups verwendet, die mit `mysqldump` erstellt wurden. Wenn Sie beim Aufruf von `mysql` keinen Datenbanknamen angeben, müssen Sie die gewünschte Datenbank mit dem Kommando `USE` einstellen.

Die folgenden Optionen betreffen lediglich den Verbindungsaufbau und gelten in gleicher Form auch für die im Folgenden beschriebenen Kommandos `mysqladmin` und `mysqldump`. In der Praxis sind zumeist nur die Optionen `-u`, `-p` und `-h` erforderlich.

► `--default-character-set=name`

gibt an, welcher Zeichensatz für die Kommunikation mit dem MySQL-Server verwendet werden soll (z. B. `latin1` oder `utf8`).

► `--defaults-extra-file=dateiname`

lädt Verbindungsoptionen aus der angegebenen Datei. Das ist beispielsweise zweckmäßig, wenn `mysql` oder `mysqldump` regelmäßig durch Cron ausgeführt werden soll und eine interaktive Passwordeingabe unmöglich ist.

► `-h` bzw. `--host=hostname`

gibt den Namen oder die IP-Nummer des Rechners an, auf dem der Server läuft (standardmäßig `localhost`).

► `--login-path=gruppe`

wertet die Daten von [gruppe] in `.mylogin.cnf` aus (ab MySQL 5.6).

► `-P` bzw. `--port=n`

gibt die Port-Nummer für die TCP/IP-Verbindung an (standardmäßig 3306).

► `-p` bzw. `--password`

fordert unmittelbar nach dem Start des Kommandos zur Eingabe des Passworts auf. Wenn die Option fehlt, liest `mysql` das Passwort aus der lokalen Konfigurations-

datei `.my.cnf`. Wenn diese Datei nicht existiert oder keine Passwortdaten enthält, versucht `mysql` den Login ohne Passwort.

- ▶ `-pxxx` bzw. `--password=xxx`

übergibt das Passwort direkt. Das ist allerdings unsicher, denn das Passwort erscheint in der Prozessliste im Klartext! Bei der Option `-p` darf im Gegensatz zu anderen Optionen kein Leerraum angegeben werden.

- ▶ `--protocol=tcp/socket`

gibt an, welches Protokoll zur Kommunikation zwischen dem MySQL-Client und dem Datenbank-Server verwendet werden soll. Standardmäßig gilt `socket`, wenn Client und Server auf demselben Rechner laufen, sonst `tcp`.

- ▶ `-S` bzw. `--socket=name`

gibt den Ort der Socket-Datei an. Diese Option ist nur erforderlich, wenn `mysql` die Socket-Datei nicht selbstständig findet. Die Datei hat bei den meisten Distributionen den Namen `/var/run/mysqld/mysqld.sock`. Ihr Ort kann durch die Konfigurationsdatei `/etc/my.cnf` bzw. `/etc/mysql/my.cnf` eingestellt werden.

- ▶ `-u` bzw. `--user=name`

gibt den MySQL-Benutzernamen an (standardmäßig den aktuellen Benutzernamen).

Außerdem gibt es einige `mysql`-spezifische Optionen:

- ▶ `-B` bzw. `--batch`

trennt bei der Ausgabe der Ergebnisse die Spalten durch Tabulatorzeichen (statt durch Leerzeichen und Liniengrafik). Außerdem werden nur die Ergebnisse von Abfragen angezeigt, aber keinerlei Statusinformationen.

- ▶ `-e` bzw. `--execute='sql-kommando'`

führt das angegebene Kommando aus. Es dürfen auch mehrere, durch Strichpunkte getrennte Kommandos angegeben werden.

- ▶ `-h` bzw. `--html`

formatiert das Abfrageergebnis als HTML-Code.

- ▶ `-r` bzw. `--raw`

gibt die Zeichen `O`, Tabulator, Newline und `\` in Abfrageergebnissen unverändert aus. (Normalerweise werden diese Zeichen als `\0`, `\t`, `\n` und `\\` ausgegeben.) Die Option ist nur in Kombination mit `--batch` wirksam.

- `-x` bzw. `--xml`

formatiert das Abfrageergebnis als HTML-Code.

### Beispiel

Das folgende Kommando spielt ein Datenbank-Backup auf dem lokalen MySQL-Server ein. Die Datenbank `dbname` muss bereits existieren.

```
user$ mysql -u name -p dbname < backup.sql
Enter password: *****
```

**mysqladmin** [optionen] kommando1 kommando2 ...

`mysqladmin` hilft bei diversen administrativen Aufgaben auf einem MySQL- oder MariaDB-Server – etwa beim Anlegen neuer Datenbanken. An `mysqladmin` können mehrere Kommandos übergeben werden, die der Reihe nach ausgeführt werden. Die Namen von `mysqladmin`-Kommandos können so weit abgekürzt werden, wie sie noch eindeutig erkennbar sind (z. B. `flush-l` statt `flush-logs`). Die meisten `mysqladmin`-Kommandos können auch als SQL-Kommandos ausgeführt werden – etwa durch `CREATE DATABASE`, `DROP DATABASE`, `FLUSH` etc.

Damit `mysqladmin` eine Verbindung zum Datenbank-Server herstellen kann, müssen die beim `mysql`-Kommando beschriebenen Verbindungsoptionen verwendet werden, also `-u`, `-p`, `-h` etc. Außerdem kennt `mysqladmin` einige weitere Optionen:

- `-f` bzw. `--force`

führt das Kommando ohne Rückfragen aus. Die Ausführung mehrerer Kommandos wird auch dann fortgesetzt, wenn bei einem Kommando ein Fehler auftritt.

- `-i` bzw. `--sleep=n`

wiederholt das Kommando alle *n* Sekunden (beispielsweise zur regelmäßigen Statusanzeige). `mysqladmin` läuft nun endlos und muss mit `[Strg]+[C]` beendet werden.

- `-r` bzw. `--relative`

zeigt in Kombination mit `-i` und dem Kommando `extended-status` die Veränderung gegenüber dem vorherigen Zustand an.

Im Folgenden sind die wichtigsten `mysqladmin`-Kommandos beschrieben:

- ▶ `create dbname`  
erzeugt eine neue Datenbank.
- ▶ `drop dbname`  
löscht eine vorhandene Datenbank unwiderruflich.
- ▶ `extended-status`  
zeigt zahllose Statusvariablen des Servers an.
- ▶ `flush-logs`  
schließt alle Logging-Dateien und öffnet sie wieder .
- ▶ `flush-privileges`  
liest die Datenbank mit den MySQL-Zugriffsrechten neu ein.
- ▶ `kill id1, id2 ...`  
beendet die angegebenen Threads.
- ▶ `ping`  
testet, ob eine Verbindung zum Datenbank-Server hergestellt werden kann.
- ▶ `shutdown`  
beendet den MySQL-Server.
- ▶ `status`  
zeigt diverse Statusvariablen des MySQL-Servers an.
- ▶ `variables`  
liefert eine Liste der Systemvariablen des MySQL-Servers.
- ▶ `version`  
ermittelt die Version des MySQL-Servers.

### Beispiel

Das folgende Kommando erzeugt eine neue Datenbank auf dem lokalen MySQL- oder MariaDB-Server:

```
user$ mysqladmin -u name -p create newedb
Enter password: *****
```

```
mysqldump [optionen] dbname [tabelle1 tabelle2 ...]
mysqldump [optionen] --databases [dbname1 dbname2 ...]
mysqldump [optionen] --all-databases
```

mysqldump ist ein einfaches Backup-Werkzeug für die Datenbanksysteme MySQL und MariaDB. Das Kommando liefert eine lange Liste aller SQL-Kommandos, aus denen die Datenbank exakt wiederhergestellt werden kann. mysqldump kennt drei Syntaxvarianten, je nachdem, ob eine Datenbank, mehrere aufgezählte Datenbanken oder alle von MySQL verwalteten Datenbanken gespeichert werden sollen. Nur bei der ersten Variante kann das Backup auf einzelne Tabellen beschränkt werden.

Damit mysqldump eine Verbindung zum Datenbank-Server herstellen kann, müssen die beim mysql-Kommando beschriebenen Verbindungsoptionen verwendet werden, also -u, -p, -h etc. Eine Menge weiterer Optionen steuern die Details des Backups.

► --add-locks

fügt vor dem ersten INSERT-Kommando LOCK TABLE und nach dem letzten INSERT-Kommando UNLOCK ein. Das beschleunigt das Wiedereinlesen von MyISAM-Tabellen. Die Option ist aber für InnoDB-Tabellen ungeeignet.

► --create-options

inkludiert MySQL-spezifische Optionen in das CREATE-TABLE-Kommando.

► --disable-keys

deaktiviert die automatische Schlüsselaktualisierung während des Wiedereinlesens der Daten. Dadurch werden Indizes erst am Ende des Einfügeprozesses aktualisiert, was schneller ist.

► --extended-insert

erzeugt INSERT-Kommandos, mit denen jeweils mehrere Datensätze gleichzeitig eingefügt werden.

► --lock-all-tables

führt für die gesamte Datenbank LOCK TABLE READ aus. Das stellt sicher, dass während des gesamten Backups keine Tabelle verändert werden kann. Diese Option ist nur bei MyISAM-Tabellen zweckmäßig. Bei InnoDB-Tabellen verwenden Sie besser --single-transaction.

► --lock-tables

führt während des Backups für jede einzelne Tabelle LOCK TABLE READ aus, bevor die Daten gelesen werden.

► `--no-create-info`

speichert nur die Daten, aber nicht die Struktur der Datenbank. Im resultierenden Backup gibt es also nur INSERT-Kommandos, aber kein CREATE TABLE.

► `--no-data`

sichert nur die Struktur (das Schema) der Datenbank, nicht aber die darin enthaltenen Daten.

► `--opt`

ist die Kurzschreibweise für die Optionen `--add-drop-table`, `--add-locks`, `--create-options`, `--disable-keys`, `--extended-insert`, `--lock-tables`, `--quick` und `--set-charset`.

Aus historischen Gründen ist `--opt` standardmäßig aktiv. Tatsächlich sind die resultierenden Einstellungen aber nur dann optimal (daher kommt der Name dieser Option!), wenn Sie ein Backup von MyISAM-Tabellen durchführen.

► `--quick`

gibt die zu sichernden Datensätze ohne Zwischenspeicherung im RAM direkt aus.

► `--routines`

sichert auch den Code von Stored Procedures.

► `--set-charset`

bewirkt, dass am Beginn der `mysqldump`-Ausgabe der aktive Zeichensatz verändert und am Ende der bisher geltende Zeichensatz wiederhergestellt wird.

► `--single-transaction`

bewirkt, dass das gesamte Backup im Rahmen einer einzigen Transaktion durchgeführt wird. Das ist bei InnoDB-Tabellen zweckmäßig und schließt aus, dass einzelne Tabellen während des Backups verändert werden. Somit wird sichergestellt, dass das Backup insgesamt konsistent ist.

► `--skip-opt`

deaktiviert die standardmäßig aktive Option `--opt`.

► `--triggers`

sichert auch den Code von Triggern.

### Beispiele

Für ein Backup einer Datenbank mit MyISAM-Tabellen sind nur wenige Optionen erforderlich. Wenn Sie ausschließen möchten, dass einzelne Tabellen wäh-



rend des Backups geändert werden, fügen Sie dem Kommando noch die Option `--lock-all-tables` hinzu. Damit sind während des Backups allerdings alle Schreiboperationen in der Datenbank blockiert.

```
user$ mysqldump -u user -p dbname > backup.sql
Enter password: *****
```

Die optimale Durchführung eines Backups mit InnoDB-Tabellen erfordert wesentlich mehr Optionen, weil die Standardeinstellungen von `mysqldump` nicht für diesen Fall optimiert sind:

```
user$ mysqldump -u user -p --skip-opt --single-transaction \
    --create-options --quick --extended-insert \
    --disable-keys --add-drop-table dbname > backup.sql
```

Wenn Sie das Backup automatisiert durch einen Cron-Job ausführen, ersetzen Sie `-u user -p` durch die folgende Option und geben die Verbindungsparameter in der nur für root lesbaren Datei `/root/.my.cnf` an:

```
# im Backup-Script
mysqldump --defaults-extra-file=/root/.my.cnf ...
```

Um das Backup später wieder einzuspielen, führen Sie diese Kommandos aus:

```
user$ mysqladmin -u user -p create dbname
user$ mysql -u user -p --default-character-set=utf8 dbname < backup.sql
```

**namei** [optionen] datei

`namei` zeigt die Zugriffsrechte aller Verzeichnisse, die den Pfad der angegebenen Datei bilden:

```
user$ namei -l ~/.ssh
f: /home/kofler/.ssh
drwxr-xr-x root root /
drwxr-xr-x root root home
drwxr-xr-x kofler kofler kofler
drwx----- kofler kofler .ssh
```

**netstat** [optionen]

`netstat` liefert Informationen über die Netzwerkaktivität auf dem lokalen Rechner. Wird das Kommando ohne Optionen aufgerufen, liefert es eine Liste aller offenen Internetverbindungen und Sockets.

### ► -a

berücksichtigt auch nichtaktive Sockets, also Dienste, die auf einem Netzwerk-Port auf eine Verbindung warten (Zustand `LISTEN` bzw. bei deutscher Lokalisierung `HÖREN`).

### ► -e

gibt bei Internetverbindungen zusätzlich den Benutzer an.

### ► -n

liefert numerische Adressen und Port-Nummern anstelle von Host- und Netzwerkdienstnamen.

### ► -p

gibt zusätzlich die Prozessnummer (PID) und den Prozessnamen des Programms an, das für die Verbindung verantwortlich ist. `netstat` benötigt root-Rechte, damit es Informationen über nichteigene Prozesse liefern kann.

### ► -t / -u / -w / -X

schränkt die Ausgabe auf Verbindungen ein, die das Protokoll TCP, UDP, Raw oder Unix nutzen.

## Beispiel

Das folgende Kommando listet alle aktiven Verbindungen (`established`) bzw. überwachten Ports (`LISTEN`) auf. Die Ausgabe wurde aus Platzgründen stark gekürzt.

```
root# netstat -atupe
```

Active Internet connections (servers and established)

Proto	Local Address	Foreign Addr	State	User	PID/Prog name
tcp	*:nfs	*:*	LISTEN	root	-
tcp	*:ldap	*:*	LISTEN	root	5842/slapd
tcp	*:3142	*:*	LISTEN	root	5904/perl
tcp	localhost:mysql	*:*	LISTEN	mysql	5785/mysqld
tcp6	[::]:ssh	[::]:*	LISTEN	root	5559/sshd
tcp6	mars.sol:ssh	merkur.so...	ESTAB	root	7729/0
udp	*:nfs	*:*		root	-
udp	mars.local:netbios-ns	*:*		root	6231/nmbd

## newaliases

Die Datei `/etc/aliases` enthält eine Alias-Liste für den E-Mail-Server, die z. B. dafür sorgt, dass alle E-Mails, die an `postmaster` adressiert sind, an `root` weitergeleitet wer-

den. Damit Änderungen an dieser Datei vom Mail-Server berücksichtigt werden, müssen Sie `newaliases` ausführen.

**newgrp** [gruppenname]

Das Kommando `newgrp` bestimmt die gerade aktive Gruppe eines Benutzers, der mehreren Gruppen angehört. Die aktive Gruppe bestimmt, welcher Gruppe neu erzeugte Dateien angehören. Die zur Auswahl stehenden Gruppen können mit `groups` ermittelt werden. Wenn kein Gruppenname angegeben wird, wird die primäre Gruppe verwendet. Diese Gruppe gilt auch nach einem Login automatisch als aktive Gruppe.

### Beispiel

Im folgenden Beispiel macht das `newgroup`-Kommando die Gruppe `docuteam` zur aktiven Gruppe des Benutzers `kofler`. Die neu erzeugte Datei `newfile` wird daher der Gruppe `docuteam` zugeordnet und kann von anderen Mitgliedern des Dokumentations-Teams bearbeitet werden.

```
user$ groups
kofler docuteam wheel
user$ newgroups docuteam
user$ touch newfile
```

**newusers** datei

`newusers` liest eine Textdatei und erzeugt für jede Zeile einen neuen Benutzer. Die Textdatei weist prinzipiell dasselbe Format auf wie `/etc/passwd`. Allerdings müssen die Passwörter unverschlüsselt in der zweiten Spalte angegeben werden. Die meisten weiteren Parameter (UID, GID etc.) sind optional. `newusers` erzeugt für jeden angegebenen Benutzer einen neuen Account, wobei bei Bedarf auch die dazugehörenden Gruppen angelegt werden. Für fehlende Parameter wählt `newusers` geeignete Defaultwerte, wobei die Einstellungen in `/etc/login.defs` berücksichtigt werden.

Beachten Sie, dass `newusers` zwar neue Heimatverzeichnisse erzeugt, wenn deren Ort in der sechsten Spalte angegeben ist. Das Kommando kümmert sich aber nicht darum, den Inhalt von `/etc/skel` dorthin zu kopieren.

### Beispiel

Die folgenden Zeilen zeigen die minimalistische Textdatei `users.txt`, die den Anforderungen von `newusers` entspricht. Da `users.txt` Klartextpasswörter enthält, müssen

Sie darauf achten, dass niemand außer `root` die Datei lesen kann, bzw. die Datei nach der Ausführung von `newusers` wieder löschen.

```
huber:geheim1::Hermann Huber:/home/huber:/bin/bash
moser:geheim2::Gabi Moser:/home/moser:/bin/bash
schmidt:geheim3::Peter Schmidt:/home/schmidt:/bin/bash
```

`newusers` erzeugt nun die drei neuen Benutzer `huber`, `moser` und `schmidt` sowie gleichnamige primäre Gruppen. `newusers` entscheidet sich selbst für geeignete UIDs und GIDs.

```
root# newusers users.txt
```

**nice** [optionen] programm

`nice` startet das angegebene Programm mit einer verringerten oder erhöhten Priorität. Das Kommando kann dazu eingesetzt werden, nicht zeitkritische Programme mit kleiner Priorität zu starten, um das restliche System nicht zu stark zu beeinträchtigen.

► `-n +/-n`

gibt den `nice`-Wert vor. Standardmäßig (also ohne `nice`) werden Programme mit dem `nice`-Wert 0 gestartet. Ein Wert von `-20` bedeutet »höchste Priorität«, ein Wert von `+19` bedeutet »niedrigste Priorität«. Werte kleiner 0 dürfen nur von `root` angegeben werden, d. h., die meisten Anwender können mit `nice` nur Programme mit reduzierter Priorität starten. Wenn auf diese Option verzichtet wird, startet `nice` das Programm mit dem `nice`-Wert von `+10`.

Beachten Sie, dass `nice` nur die CPU-Lastung steuert. Wenn Sie die IO-Lastung eines Kommandos reduzieren möchten, setzen Sie besser `ionice` ein.

### Beispiel

Das folgende Kommando startet das Backup-Skript `sichere` mit niedrigerer Priorität:

```
user$ nice -n 10 sichere
```

**nl** [optionen] datei

`nl` nummeriert alle nichtleeren Zeilen der angegebenen Textdatei und schreibt das Ergebnis in die Standardausgabe. Durch die Einstellung der zahlreichen Optionen kann eine seitenweise Nummerierung, eine Nummerierung von Kopf- und Fußzeilen etc. erreicht werden.

```
nmmap [optionen] rechnername/ip-adresse
```

Das Kommando `nmmap` aus dem gleichnamigen Paket führt auf dem angegebenen Rechner einen sogenannten Port-Scan durch und versucht festzustellen, welche Netzwerkdienste aktiv sind. `nmmap` sollte ausschließlich zur Analyse eigener Rechner eingesetzt werden. Ein Port-Scan fremder Rechner kann als Einbruchversuch gewertet werden!

`nmmap` wird durch unzählige Optionen gesteuert, deren Beschreibung hier den Rahmen sprengen würde. In vielen Fällen reicht `nmmap -v -A name`, um einen ersten Überblick über die Netzwerkdienste des angegebenen Rechners zu bekommen. Fortgeschrittene `nmmap`-Anwender finden weitere Details in der umfassenden `man`-Seite oder auf <http://insecure.org/nmap>. Zu `nmmap` existiert auch eine grafische Benutzeroberfläche, die sich üblicherweise im Paket `nmmap-frontend` befindet.

### Beispiel

Die folgenden Zeilen zeigen das stark gekürzte `nmmap`-Ergebnis für einen im lokalen Netz laufenden Rechner, auf dem ein SSH- und ein Samba-Server laufen:

```
root# nmmap -v -A mars
Scanning mars (10.0.0.11) [1 port]
...
Discovered open port 445/tcp on 10.0.0.11
Discovered open port 22/tcp on 10.0.0.11
Discovered open port 139/tcp on 10.0.0.11
...
Host script results:
| nbstat:
|   NetBIOS name: LOCALHOST, NetBIOS user: <unknown>, NetBIOS MAC: <unknown>
| smb-os-discovery:
|   OS: Unix (Samba 3.2.15)
| smb-security-mode:
|   Account that was used for smb scripts: guest
|   User-level authentication
|_smbv2-enabled: Server doesn't support SMBv2 protocol
...
Nmap done: 1 IP address (1 host up) scanned in 19.40 seconds
      Raw packets sent: 1907 (84.702KB) | Rcvd: 1026 (41.582KB)
```

```
nmcli [optionen] con|dev|nm kommando
```

Der Network Manager wird üblicherweise durch ein Menü im Gnome- oder KDE-Panel gesteuert. Mit dem Kommando `nmcli` können Sie Netzwerkverbindungen auch über die Kommandozeile bzw. durch ein Script steuern.

Die zur Auswahl stehenden Kommandos hängen davon ab, auf welches Objekt sie sich beziehen. Mit den Optionen `-t`, `-p` etc. kann die Form der `nmcli`-Ausgaben gesteuert werden – je nachdem, ob die Ausgabe ordentlich formatiert oder durch ein Script weiterverarbeitet werden soll (Details siehe `man nmcli`).

- ▶ `con down name` bzw. `con down uuid n`  
deaktiviert die angegebene Verbindung.
- ▶ `con show`  
listet alle konfigurierten Verbindungen auf, wobei zu jeder Verbindung ihr Name und ihre UUID angegeben wird.
- ▶ `con up name` bzw. `con up uuid n`  
aktiviert die angegebene Verbindung.
- ▶ `dev disconnect name`  
beendet die Verbindung für die angegebene Schnittstelle.
- ▶ `dev list`  
listet alle dem Network Manager bekannten Netzwerkschnittstellen auf und gibt deren Eigenschaften an.
- ▶ `dev show`  
zeigt ausführliche Informationen zu allen Netzwerkschnittstellen an.
- ▶ `dev wifi list`  
liefert eine Liste aller in Reichweite befindlichen WLAN-Netzwerke.
- ▶ `general [status]`  
zeigt den Status des Network Managers an.
- ▶ `networking [on|off]`  
zeigt an, ob eine Netzwerkverbindung besteht oder nicht. Durch `off` bzw. `on` können alle Netzwerkverbindungen unterbrochen bzw. wiederhergestellt werden.
- ▶ `radio [wifi|wwan|wimax|all] [on|off]`  
zeigt den Status der Funknetzwerke an bzw. verändert deren Status. `wifi` bezeichnet WLAN-Verbindungen, `wwan` Mobilfunkverbindungen und `wimax` die bei uns unübliche WiMAX-Technik.

## Beispiel

Das erste Kommando listet zuerst alle dem Network Manager bekannten Verbindungen auf. Die folgenden beiden Kommandos de- und reaktivieren die Verbindung mit dem Namen *Kabelnetzwerkverbindung 1*:

```
root# nmcli con show
NAME                                UUID                                TYP                                GERÄT
Kabelnetzwerkverbindung 1          0ffd094d-...                        802-3-ethernet                    enp0s3
root# nmcli con down id 'Kabelnetzwerkverbindung 1'
root# nmcli con up id 'Kabelnetzwerkverbindung 1'
```

Das zweite Beispiel filtert aus den detaillierten Schnittstelleninformationen die IP-Adressen des Nameservers heraus. Das ist vor allem unter Ubuntu praktisch, wo diese Adressen nicht aus `/etc/resolv.conf` hervorgehen. Vielmehr verweist `resolv.conf` bei dieser Distribution auf einen lokalen Nameserver, also auf die Adresse 127.0.0.1.

```
root# nmcli dev show | grep DNS
IP4.DNS[1]: 10.0.0.138
IP4.DNS[2]: 8.8.8.8
```

## nohup kommando

Wenn Sie ein Kommando als Hintergrundprozess in einem Shell-Fenster starten und das Fenster dann schließen oder wenn Sie das Kommando in einer Textkonsole starten und sich dann ausloggen, wird der Hintergrundprozess automatisch beendet. Das ist im Regelfall ein sinnvolles Verhalten.

Manchmal möchten Sie aber einen Prozess starten, der nach dem Ausloggen weiterläuft – und genau dazu dient `nohup`. Das Kommando muss mit seinem vollständigen Pfad angegeben werden. Es kann keine Textausgaben an die Standardausgabe schreiben. Gegebenenfalls werden derartige Ausgaben in die Datei `nohup.out` im lokalen Verzeichnis umgeleitet.

## Beispiel

Im folgenden Beispiel loggt sich ein Administrator via `ssh` auf einem Server ein, startet dort ein Backup-Skript im Hintergrund und loggt sich dann wieder aus. Das Backup-Skript läuft weiter.

```
userx@localhost$ ssh usery@remotehost
usery@remotehost$ nohup backup-script &
usery@remotehost$ exit
```

**nvidia-xconfig** [optionen]

nvidia-xconfig hilft dabei, den proprietären NVIDIA-Grafiktreiber zu konfigurieren. Das Kommando steht nur zur Verfügung, wenn dieser Treiber installiert ist.

Wenn das Kommando ohne Parameter ausgeführt wird, erzeugt bzw. verändert es die Datei `/etc/X11/xorg.conf` so, dass in Zukunft der NVIDIA-Grafiktreiber verwendet wird. Wenn nach einem Neustart des Rechners das Grafiksystem funktioniert, kann zur weiteren Konfiguration die grafische Benutzeroberfläche `nvidia-settings` verwendet werden.

- ▶ `-c datei`  
verwendet die angegebene Datei anstelle von `/etc/X11/xorg.conf`.
- ▶ `--mode=WxH`  
fügt `xorg.conf` einen Grafikmodus für  $W \times H$  Pixel hinzu.
- ▶ `--query-gpu-Infos`  
zeigt Details zu allen vorhandenen Grafikkarten und angeschlossenen Monitoren an.
- ▶ `-t`  
liest `xorg.conf` und zeigt die darin enthaltenen Einstellungen in einer übersichtlichen Baumdarstellung an.

**openssl** kommando

openssl aus dem gleichnamigen Paket hilft beim Erzeugen und der weiteren Administration von Zertifikaten sowie privaten und öffentlichen Schlüsseln. Das Kommando greift dabei auf die OpenSSL-Bibliothek zurück. Neue Zertifikate oder Schlüssel werden an die Standardausgabe geschrieben, wenn Sie nicht mit `-out` eine Zielfile angeben.

- ▶ `genkey` [optionen]  
erzeugt einen privaten Schlüssel. Um einen 2048-Bit-RSA-Schlüssel in eine Datei zu schreiben, sind die folgenden Optionen erforderlich:

```
openssl genkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out server.key
```

Wenn der Schlüssel selbst verschlüsselt werden soll, geben Sie den Namen des Verschlüsselungsalgorithmus durch eine weitere Option an, z. B. `-aes256`. Bei der Ausführung fragt openssl nun zweimal nach einem Verschlüsselungspasswort.



- ▶ `list-standard-commands | list-message-digest-commands |`  
`list-cipher-commands | list-cipher-algorithms |`  
`list-message-digest-algorithms | list-public-key-algorithms`

listet die von openssl unterstützten Kommandos, Algorithmen etc. auf.

- ▶ `req [optionen]`

erzeugt eine Anfrage zur Signierung eines Zertifikats (also ein *Certificate Signing Request*, eine CSR-Datei). Mit `-new` geben Sie an, dass ein neues Zertifikat eingerichtet werden soll. `-key` gibt die dabei zu verwendende Schlüsseldatei an. Ohne diese Option erzeugt openssl zu diesem Zweck einen neuen RSA-Schlüssel, der mit `-keyout` in einer eigenen Datei gespeichert werden kann.

```
openssl req -new -sha256 -key server.key -out server.csr
```

openssl fragt dann interaktiv nach den Eckdaten des Zertifikats, also nach dem Ländercode, dem Ort, Ihrem Namen, dem Hostnamen (Common Name) etc.

Mit den zusätzlichen Optionen `-x509 -days n` erzeugt openssl ein selbst signiertes Zertifikat mit einer Gültigkeit von `n` Tagen anstelle einer Zertifikatsanfrage.

- ▶ `rsa [optionen]`

verarbeitet RSA-Schlüssel und konvertiert sie zwischen unterschiedlichen Formen und Formaten. `-in datei` und `-out datei` geben an, aus welcher Datei der ursprüngliche Schlüssel gelesen und wohin der neue Schlüssel geschrieben wird.

Werden keine weiteren Optionen angegeben, entfernt openssl die Verschlüsselung (*pass phrase*) des Schlüssels. Mit `-inform` und `-outform` kann die Form des Schlüssels spezifiziert werden (standardmäßig PEM, alternativ DER oder NET). Mit `-in datei -text` können Sie die Eigenschaften eines vorhandenen Schlüssels ermitteln, z. B. die Schlüssellänge (die Anzahl der Bits).

- ▶ `x509 [optionen]`

zeigt Zertifikatsdaten an, signiert Zertifikate und konvertiert Zertifikate zwischen verschiedenen Formen. `-in` und `-out` geben wieder die Dateinamen der Quell- und Zielzertifikate an. Mit `-req` erwartet openssl eine Zertifikatsanfrage als Eingabe, nicht das Zertifikat selbst. `-CA` gibt an, welches CA-Zertifikat für die Signatur verwendet werden soll. `-CAkey` gibt den dazugehörenden privaten Schlüssel an.

Zu den meisten openssl-Kommandos gibt es eigene `man`-Seiten. Beispielsweise beschreibt `man req` die unzähligen Optionen von `openssl req`, `man rsa` beschreibt die Optionen von `openssl rsa` etc.

### Beispiele

Wenn Sie für Ihren Webserver ein Zertifikat von einer offiziellen Zertifizierungsstelle (z. B. Thawte) unterzeichnen lassen möchten, erzeugen Sie zuerst einen Schlüssel (wenn Sie noch keinen haben) und dann ein Certificate Signing Request. Auf eine Verschlüsselung des Schlüssels wird hier verzichtet, weil der Webserver sonst bei jedem Start nach dem Verschlüsselungspasswort fragen müsste.

```
root# genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 -out server.key
root# chmod 400 server.key
root# openssl req -new -key server.key -out server.csr
...
Common Name (eg server FQDN or YOUR name) []: firma-abc.de
```

Zur Kontrolle können Sie einen Blick in die CSR-Datei werfen:

```
root# openssl req -in server.csr -noout -text
```

Die CSR-Datei übermitteln Sie nun an die Zertifizierungsstelle. Von dort bekommen Sie dann gegen gutes Geld das signierte Zertifikat. Die übliche Dateikennung lautet .pem oder .crt. Diese Datei, Ihren eigenen Schlüssel sowie ein CA-Zertifikat der Zertifizierungsstelle müssen Sie dann bei der Apache-Konfiguration angeben.

```
# Apache-Konfigurationsdatei
SSLCertificateFile      /etc/apache2/server.pem
SSLCertificateKeyFile   /etc/apache2/server.key
SSLCertificateChainFile /etc/apache2/sub.class1.server.ca.pem
SSLCACertificateFile    /etc/apache2/ca.pem
```

Wenn Sie die Kosten für eine offizielle Signatur scheuen, können Sie sich Ihr Zertifikat selbst signieren:

```
root# openssl x509 -req -days 1900 -in server.csr \
      -signkey server.key -sha256 -out server.pem
Signature ok
subject=C=DE/L=Berlin/O=Max Muster/CN=www.firma-abc.de/
      emailAddress=webmaster@firma-abc.de
Getting Private key
```

Um einen Blick in die Zertifikatsdatei zu werfen, führen Sie `openssl x509 -text` aus:

```
root# openssl x509 -text -in server.pem
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 12669601459972319941 (0xafd37766c36baac5)
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=DE, L=Berlin, O=Max Muster,
                CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de
```

```

Validity
    Not Before: Sep 28 14:48:03 2015 GMT
    Not After : Dec 10 14:48:03 2020 GMT
...

```

Das Erzeugen eines Schlüssels, einer Zertifikatsanfrage und das Selbst-Signieren kann auch in einem einzigen Kommando erledigt werden: Das folgende Kommando erzeugt ein für 10 Jahre gültiges, selbst signiertes Zertifikat für einen Mail-Server. Entscheidend ist, dass Sie bei der Ausführung von `openssl` als Common Name den Hostnamen Ihres Servers angeben. Da das Zertifikat selbst signiert ist, wird Ihr Mail-Client bei der Konfiguration darauf hinweisen, dass das Zertifikat nicht vertrauenswürdig ist.

```

root# openssl req -new -x509 -days 3650 -nodes \
    -out /etc/ssl/certs/postfix.pem \
    -keyout /etc/ssl/private/postfix.key
...
Common Name (eg server FQDN or YOUR name) []: firma-abc.de
root# chmod 400 /etc/ssl/private/postfix.key

```

**pactl** kommando [parameter]

`pactl` steuert das PulseAudio-System. Dieser netzwerkfähige Sound-Server ist auf den meisten Linux-Distributionen standardmäßig installiert und verwendet zumeist ALSA als Basis. PulseAudio macht es möglich, dass mehrere Programme unkompliziert parallel das Audio-System nutzen können. `pactl` kennt rund 30 Kommandos, von denen hier nur die wichtigsten präsentiert werden:

- ▶ `info|stat`  
zeigt die Eckdaten bzw. den Speicherverbrauch des PulseAudio-Systems an.
- ▶ `list [short] [modules|sinks|sources|clients|cards|...]`  
liefert eine detaillierte Beschreibung aller Module, Audio-Quellen und -Ausgänge, Streams etc., die von PulseAudio verwaltet werden. Das zusätzliche Schlüsselwort `short` reduziert die Ausgabe auf eine Zeile pro Element. Die Angabe von `modules`, `sinks` bewirkt, dass `pactl` nur die betreffenden Elemente auflistet.
- ▶ `set-default-sink n`  
legt den Default-Audio-Ausgang fest. Anstelle der Nummer des Audio-Systems `n` (siehe `/proc/asound/cards`) können Sie auch dessen Namen angeben. Die zulässigen Namen ermitteln Sie mit `pactl list short cards`.

- ▶ `set-sink-mute n 0|1|toggle`  
deaktiviert, aktiviert bzw. verändert die Mute-Funktion für den Audio-Ausgang *n*.
- ▶ `set-sink-volume n vol`  
stellt die Lautstärke für den Audio-Ausgang *n* ein. Die Lautstärke kann als Integerzahl, als Prozentwert oder als Dezibel-Wert angegeben werden (z. B. 10% oder 20dB).

### Beispiel

Die beiden folgenden Kommandos führen dazu, dass das erste Audio-Gerät als Standardausgabegerät bei einer Lautstärke von 20 % verwendet wird:

```
user$ pactl set-default-sink 0
user$ pactl set-sink-volume 20%
```

```
pandoc [optionen] in1 [in2 in3 ...] [> out]
```

Die primäre Aufgabe von `pandoc` aus dem gleichnamigen Paket besteht darin, Markdown-Textdateien in andere Formate umzuwandeln, beispielsweise in HTML-Dateien, PDF-Dokumente oder Office-Dateien. Mit Einschränkungen kann `pandoc` anstelle von Markdown-Dateien auch andere Eingabeformate verarbeiten, unter anderem HTML und LaTeX.

- ▶ `-c cssdatei`  
bettet die angegebene CSS-Datei in das HTML-Dokument ein. Die Option kann mehrfach angegeben werden, wenn mehrere CSS-Dateien erforderlich sind.
- ▶ `-f format`  
gibt das Quellformat an (*from*). Wichtige Quellformate sind `asciidoc`, `docbook`, `latex` und `markdown`. `pandoc` versucht anhand der Dateikennung der Ein- und Ausgabedateien die gewünschten Quell- und Zielformate selbst zu erkennen. Die Optionen `-f` und `-t` sind nur erforderlich, wenn dies nicht gelingt.
- ▶ `-H headerdatei`  
fügt die angegebene Header-Datei am Beginn des HTML- oder LaTeX-Quelltexts ein. Die Option kann mehrfach angegeben werden, wenn mehrere Header-Dateien erforderlich sind.
- ▶ `-o output.kennung`  
schreibt das Ergebnis in die angegebene Datei anstatt zur Standardausgabe. Die Option ist bei binären Ausgabeformaten (DOCX, ODT, PDF) zwingend erforderlich.

► -s

erzeugt ein eigenständiges Dokument (*standalone*). Diese Option ist vor allem bei den Ausgabeformaten HTML, LaTeX und RTF relevant. Ohne diese Option erzeugt pandoc Dateien ohne Header-Informationen; solche Dateien können nicht für sich verwendet werden, sondern müssen in ein anderes Dokument eingebettet werden.

► -t format

gibt das Zielformat an (*to*). Wichtige Zielformate sind docx, latex, html, html5, man, odt und rtf. Beachten Sie, dass pdf nicht zur Liste der unterstützten Formate zählt. Um PDF-Dokumente zu erzeugen, müssen Sie die Option -o out.pdf angeben. pandoc wandelt den Quelltext dann zuerst in das LaTeX-Format um und erzeugt anschließend die PDF-Datei mit pdflatex.

► --toc

bettet ein Inhaltsverzeichnis in das resultierende Dokument ein.

## Beispiele

Das erste Kommando erzeugt aus der Markdown-Datei input.text ein eigenständiges HTML-Dokument. Das zweite Kommando erzeugt eine EPUB-Datei aus einem Buch, dessen Kapitel in eigenen Dateien gespeichert sind. Das dritte Kommando erzeugt eine PDF-Datei, wobei hinter den Kulissen LaTeX zum Einsatz kommt.

```
user$ pandoc -s -c mein.css input.text > output.html
user$ pandoc kap1.text kap2.text kap3.text -t epub3 -o buch.epub
user$ pandoc vorwort.text kap01.text kap02.text -o out.pdf
```

```
paplay [optionen] [datei]
parecord [optionen] [datei]
```

paplay spielt die angegebene Audio-Datei über das PulseAudio-System ab. parecord führt eine Aufnahme durch und speichert den Audio-Stream als Datei.

► -d id

bezeichnet das Ausgabegerät (paplay) bzw. die Audio-Quelle (parecord). Ohne diese Option wird das jeweilige Default-Gerät verwendet.

► --file-format=name

gibt das gewünschte Audio-Format an, z. B. raw oder wav.

► `--list-file-formats`

listet alle unterstützten Audio-Formate auf.

► `--volume=n`

gibt die gewünschte Wiedergabelautstärke an. Der Wertebereich geht von 0 bis 65.535 für maximale Lautstärke.

### Beispiel

Das folgende Kommando spielt eine der Audio-Dateien ab, die mit LibreOffice mitgeliefert werden:

```
user$ paplay /usr/lib/libreoffice/share/gallery/sounds/train.wav
```

```
paps [optionen] textdateien > out.ps
```

`paps` konvertiert die angegebenen UTF-8-Textdateien in das PostScript-Format und schreibt das Ergebnis in die Standardausgabe. Die UTF-8-Zeichen werden nicht als Buchstaben, sondern durch Linienzüge dargestellt. Aus diesem Grund sieht die Darstellung der PostScript-Datei in einem PostScript-Viewer (Evince, Okular etc.) unter Umständen pixelig aus. Eine Textauswahl ist nicht möglich. Die Druckqualität ist aber gut.

► `--columns=n`

bestimmt die Anzahl der Textspalten (standardmäßig eine).

► `--cpi=n`

steuert die Anzahl der Zeichen pro Inch (2,54 cm) und somit die Schriftgröße.

► `--font=name`

gibt den gewünschten Zeichensatz an (standardmäßig Monospace 12).

► `--landscape`

formatiert den Text im Querformat.

### Beispiel

Das folgende Kommando produziert die zweispaltige PostScript-Datei `text.ps`:

```
user$ paps --landscape --columns 2 text.txt > druck.ps
```

```
parted [optionen] [device [kommando [optionen]]]
```

parted hilft bei der Partitionierung von Festplatten und SSDs. Im Gegensatz zu fdisk kommt parted auch mit GUID-Partitionstabellen (GPT) zurecht.

Wenn Sie an parted kein Kommando übergeben, können Sie das Programm interaktiv verwenden und der Reihe nach mehrere Kommandos eingeben. Die Kommandos dürfen abgekürzt werden, sofern die Eingabe eindeutig ist (also `q` statt `quit`).

Beachten Sie dabei aber, dass die Kommandos anders als bei `fdisk` sofort ausgeführt werden und nicht mehr rückgängig gemacht werden können! Anstelle des wenig komfortablen Programms parted können Sie auch dessen grafische Benutzeroberfläche `gparted` einsetzen.

► `-a` bzw. `--align`

gibt an, wie neue Partitionen ausgerichtet werden sollen. Zulässige Einstellungen sind `none`, `cylinder` (Ausrichtung an Zylindergrenzen), `minimal` (Ausrichtung an Blockgrenzen) oder `optimal` (Ausrichtung an Vielfachen von einem binären MByte). Sie sollten unbedingt die Option `-a optimal` verwenden, um den Datenträger in maximaler Geschwindigkeit zu nutzen.

► `-l` bzw. `--list`

listet alle Partitionen auf.

► `-s` bzw. `--script`

führt alle Kommandos ohne Rückfragen aus.

Im Folgenden sind die wichtigsten parted-Kommandos kurz beschrieben:

► `align-check min/opt nr`

testet, ob die angegebene Partition an einer optimalen Position startet. `parted align-check min` testet lediglich, ob die Partition an einer physikalischen Blockgrenze startet. Bei `parted align-check opt` versucht parted aus den Eckdaten des Datenträgers die optimale Startposition von Partitionen auszurechnen. Gelingt dies nicht, gelten Vielfache von einem MByte (binär) als optimal.

► `mklabel msdos/gpt`

richtet eine neue Partitionstabelle im MBR- oder GPT-Format ein. Vorsicht: Dabei geht der gesamte Inhalt der Festplatte verloren! parted unterstützt auch diverse andere Partitionsformate, die aber im Linux-Umfeld selten von Bedeutung sind (siehe `man parted`).

- ▶ `mkpart primary/logical/extended/name [dateisys] start ende`

richtet eine neue Partition ein. Beachten Sie, dass Sie die Start- und Endposition angeben müssen, nicht die Startposition und Größe! Die Endposition kann auch als negative Zahl angegeben werden und wird dann vom Ende des Datenträgers gerechnet.

Die Angabe der Partitionsart (`primary`, `logical` oder `extended`) ist nur bei Datenträgern mit MBR-Partitionstabellen zweckmäßig. Wenn sich auf dem Datenträger eine GUID-Partitionstabelle (GPT) befindet, geben Sie der Partition mit dem ersten `mkpart`-Parameter einen Namen.

Die optionale Angabe des Dateisystems dient nur zur Festlegung des Partitions-typs. Auf der neuen Partition wird aber kein Dateisystem eingerichtet. Zulässige Typen sind `fat16`, `fat32`, `ext2`, `HFS`, `linux-swap`, `NTFS` und `ufs`.

- ▶ `print`

listet alle Partitionen auf.

- ▶ `quit`

beendet parted.

- ▶ `resize nr start ende`

verändert die Größe der angegebenen Partition. Das Kommando kann leider nur für die erweiterte Partition eines MBR-Datenträgers verwendet werden. Um die Größe anderer Partitionen zu verändern, müssen Sie diese löschen und dann mit dem exakt gleichen Startpunkt neu einrichten. Das ist natürlich eine fehleranfällige Operation, die zudem voraussetzt, dass hinter der Partition freier Platz ist.

- ▶ `rm nr`

löscht die angegebene Partition.

- ▶ `set nr flag on/off`

verändert die Flags (Zusatzattribute) der Partition. `parted` kennt unter anderem die Flags `boot`, `root`, `swap`, `hidden`, `raid`, `lvm`, `lba` und `bios_grub`.

- ▶ `unit einheit`

legt die Einheit für Positions- und Größenangaben fest. Zur Auswahl stehen:

- `s` (Sektoren)
- `B` (Byte)
- `kB`, `MB`, `GB` und `TB` ( $10^3$ ,  $10^6$ ,  $10^9$  und  $10^{12}$  Byte)



- KiB, MiB, GiB und TiB (binäre kByte, MByte, GByte und TByte)
- % (Prozent, relativ zur Gesamtgröße des Datenträgers)
- compact (dezimale MByte bei Eingaben, lesefreundliche Ausgabe)

Standardmäßig gilt die Einstellung compact. Wenn parted mit der Option --align gestartet wurde, werden beim Einrichten neuer Partitionen die Positions- und Größenangaben entsprechend angepasst.

## Beispiele

Die folgenden Zeilen zeigen, wie Sie mit parted eine neue LVM-Partition auf einer SSD mit GPT einrichten:

```
root# parted -a optimal /dev/sda
(parted) print
Modell: ATA SAMSUNG SSD 830 (scsi)
Festplatte /dev/sda: 512GB
Sektorgröße (logisch/physisch): 512B/512B
Partitionstabelle: gpt
Nummer  Anfang  Ende    Größe  Dateisys  Name                      Flags
1       17,4kB  1000MB  1000MB  fat32     EFI System Part...       boot, versteckt
2       1000MB  21,0GB  20,0GB  ext4                      versteckt
3       21,0GB  21,1GB  134MB   Microsoft reser...       versteckt, msftres
4       21,1GB  61,2GB  40,1GB  ntfs      Basic data part...       versteckt
5       61,2GB  82,2GB  21,0GB  ext4                      versteckt
6       82,2GB  103GB   21,0GB  ext4                      boot, versteckt
7       103GB   124GB   21,0GB  ext4                      boot
8       124GB   281GB   157GB   ext4
9       281GB   386GB   105GB
10      386GB   407GB   21,0GB  ext4                      lvm
(parted) mkpart lvm-partition 407GB 450GB
(parted) set 11 lvm on
(parted) print
...
11      407GB   450GB   42.7GB          lvm-partition          lvm
(parted) quit
```

parted kann Kommandos auch direkt verarbeiten, ohne manuelle Interaktion. Die folgenden Kommandos setzen voraus, dass /dev/sdb das Device eines USB-Sticks oder einer SD-Karte ist. Das erste parted-Kommando richtet dort eine neue MBR-Partitionstabelle ein und löscht damit alle vorhandenen Daten. Das zweite Kommando erzeugt eine primäre Partition, wobei am Beginn und am Ende des Datenträgers jeweils ein MByte frei bleibt. mkfs.vfat richtet in dieser Partition ein VFAT-Dateisystem ein.

```
root# parted /dev/sdb mklabel msdos
Warnung: Die bestehende Partitionstabelle und alle Daten
auf /dev/sdb werden gelöscht. Wollen Sie fortfahren?
```

```
Ja/Yes/Nein/No? ja
root# parted /dev/sdb 'mkpart primary fat32 1mib -1mib'
root# mkfs.vfat -F 32 -n FOTOS /dev/sdb1
```

**partprobe** [optionen] [devices]

**partprobe** wird üblicherweise ohne Optionen oder andere Parameter nach der Veränderung der Partitionierung lokaler Festplatten oder SSDs ausgeführt, z. B. nachdem **fdisk** beendet wurde. Das Kommando informiert nun den Kernel über die durchgeführten Änderungen, sodass die neuen bzw. veränderten Partitionen ohne einen Neustart des Rechners verwendet werden können.

**partx** [optionen] [partition] [disk]

**partx** ist ein Low-Level-Kommando, das Informationen über die Partitionierung von Datenträgern ermittelt bzw. an den Kernel weitergibt. **partx** ist gleichermaßen für die Partitionstabellen MBR und GPT geeignet.

Die zu bearbeitende Partition wird normalerweise direkt durch eine Device-Datei angegeben. Alternativ kann der Datenträger durch seine Device-Datei und die Partitionsnummer durch die Option **-n** spezifiziert werden, z. B. mit **partx -n 3 /dev/sda**. Falls eine einzelne Partition selbst wieder Sub-Partitionen enthält und daher wie ein selbstständiger Datenträger behandelt werden soll, stellen Sie dem Partitions-Device ein freistehendes Minuszeichen voran, also z. B. **partx - /dev/sda4**.

Anders als **fdisk** und **parted** verändert **partx** den Datenträger nicht, eignet sich also *nicht* zum Einrichten oder Löschen physikalischer Partitionen! Die Optionen **-a**, **-d** und **-u** verändern lediglich die Partitionstabelle des Linux-Kernels, belassen den Datenträger aber unverändert. Der Zweck dieser Optionen besteht ausschließlich darin, dem Kernel Informationen über eine mit einem anderen Programm veränderte Partitionierung mitzuteilen. Einfacher gelingt dies meist mit dem Kommando **partprobe**. In der Praxis wird **partx** oft zur Bearbeitung virtueller Datenträger verwendet, also z. B. für Loop-Devices oder für Image-Dateien virtueller Maschinen.

► **-a**

fügt neue Partitionen hinzu. Die Partition muss bereits physikalisch auf dem Datenträger existieren. **-a** aktualisiert lediglich die Partitionstabelle des Kernels!

► **-b**

verarbeitet Größenangaben in Byte.

► -d

löscht Partitionen aus der Partitionstabelle des Kernels.

► -n M:N

gibt die zu bearbeitenden Partitionen des Datenträgers numerisch an. Dabei sind verschiedene Schreibweisen möglich, wie die folgenden Beispiele verdeutlichen:

-n 4 bezeichnet die vierte Partition.

-n -2 bezeichnet die vorletzte Partition.

-n 3:6 meint die Partitionen 3 bis 6.

-n 3: bezieht sich auf alle Partitionen, beginnend mit der dritten.

-n :5 umfasst die Partitionen 1 bis 5.

► -o spalten bzw. --output spalten

zeigt ausgewählte Informationen zum ausgewählten Datenträger bzw. zur Partition an. spalten kann unter anderem die folgenden Schlüsselwörter enthalten: NR (Partitionsnummer), START (Startsektor), END (Endsektor), SIZE (Größe in Byte), TYPE (Partitionstyp als Hexcode oder UUID) und FLAGS (Zusatzinformationen). `partx -o NR,START,END` liefert somit eine Liste der Partitionen jeweils mit ihren Start- und Endsektoren. Als Sektorgröße gilt generell 512 Byte, unabhängig von der tatsächlichen Sektorgröße des Datenträgers.

► -s bzw. --show

zeigt Informationen zum ausgewählten Datenträger bzw. zur Partition an. Die mit -s oder -o angezeigten Daten werden vom physikalischen Datenträger ermittelt und stimmen unter Umständen nicht mit der Partitionstabelle des Linux-Kernels überein.

► -u

verändert die Größe oder andere Daten einer Partition in der Partitionstabelle des Kernels. Die Änderungen müssen vorher physikalisch auf dem Datenträger durchgeführt werden.

## Beispiel

Das folgende Kommando liefert Detailinformationen zu einer Festplatte mit einer MBR-Partitionstabelle:

```
root# partx -o NR,TYPE,FLAGS,START,END,SECTORS,SIZE /dev/sda
NR TYPE FLAGS  START    END  SECTORS  SIZE
 1 0x83 0x80    2048   411647  409600  200M
 2 0x8e 0x0    411648 31457279 31045632 14,8G
```

```
passwd [optionen] [username]
```

`passwd` ohne Parameter ermöglicht es, das Passwort des aktuellen Benutzers zu ändern. Dazu muss zuerst das alte und dann zweimal hintereinander das neue Passwort eingegeben werden. Das neue Passwort wird in verschlüsselter Form in die Datei `/etc/shadow` eingetragen. Allerdings muss das neue Passwort bei manchen Distributionen bestimmten Sicherheitsregeln entsprechen, damit es akzeptiert wird.

`root` kann mit `passwd name` auch das Passwort anderer Benutzer verändern. Das alte Passwort muss nicht angegeben werden, d. h., `root` kann das Passwort auch dann verändern, wenn der Benutzer sein Passwort vergessen hat. Für `root` gelten die obigen Passwortrestriktionen nicht, er kann also auch ein Passwort aus nur einem einzigen Zeichen definieren. Es ist aber nicht einmal `root` gestattet, überhaupt kein Passwort (also einfach `[ ]`) anzugeben.

Durch die Angabe von Optionen kann `passwd` einzelne Accounts auch sperren, wieder aktivieren sowie Ablaufzeiten für das Konto bzw. sein Passwort einstellen (siehe auch die Beschreibung des Kommandos `chage`, das in dieser Hinsicht noch mehr Einstellmöglichkeiten bietet).

► `-g`

stellt das Passwort einer Gruppe ein. Diese Funktion wird nur selten genutzt. Sie können damit das Betreten nichtprimärer Gruppen durch `newgrp` mithilfe eines Passworts absichern. Da das Gruppenpasswort mehreren Benutzern bekannt sein muss, ist das Verfahren inhärent unsicher.

► `-l`

deaktiviert das Konto (*lock*).

► `-u`

aktiviert ein deaktiviertes Konto wieder (*unlock*).

► `-x n`

gibt an, wie viele Tage ein Passwort maximal gültig bleibt. `-x 180` bewirkt also, dass circa alle sechs Monate ein neues Passwort eingestellt werden muss.

## Beispiel

Im folgenden Beispiel stellt `root` ein neues Passwort für den Benutzer `huber` ein. Dieser muss in Zukunft sein Passwort einmal jährlich ändern.

```
root# passwd huber
ändere Passwort für Benutzer huber.
Geben Sie ein neues Passwort ein: ******
```

```
Geben Sie das neue Passwort erneut ein:  *****
passwd: alle Authentifizierungsmerkmale erfolgreich aktualisiert.
root# passwd -x 365 huber
```

```
paste datei1 datei2 ...
```

paste setzt die Zeilen der angegebenen Dateien zu neuen (längeren) Zeilen zusammen und gibt das Ergebnis auf dem Bildschirm aus. Die erste Zeile des resultierenden Textes ergibt sich also aus der ersten Zeile der ersten Datei plus der ersten Zeile der zweiten Datei etc. Zwischen den Bestandteilen der neuen Zeile werden Tabulator-Zeichen eingefügt. Durch `> zieldatei` kann das Ergebnis in einer Datei gespeichert werden.

```
patch [optionen] < patchdatei
```

patch wendet die in einer diff-Datei zusammengefassten Änderungen an. Das Kommando wird in der Regel dazu genutzt, Codeänderungen anzuwenden (beispielsweise für den Kernelcode).

► -b

erzeugt Backup-Dateien für alle geänderten Dateien.

► --dry-run

testet den Patch, führt aber keine Änderungen aus. Generell sollten Sie vor der Anwendung jedes Patches mit der Option `--dry-run` sicherstellen, dass dabei keine Probleme auftreten. Nichts ist ärgerlicher als ein fehlerhaft oder nur teilweise angewendeter Patch!

► -pn

entfernt *n* Verzeichnisebenen aus den Dateinamen der Patch-Datei. Wenn `/a/b/name.c` der ursprüngliche Dateiname ist, macht `-p1` daraus `a/b/name.c`, `-p2` liefert `b/name.c`. Der richtige Wert *n* hängt davon ab, in welchem Verzeichnis patch ausgeführt wird.

► -R bzw. --reverse

wendet den Patch invers an. Auf diese Weise wird ein bereits angewendeter Patch wieder rückgängig gemacht.

## Beispiel

Die folgenden Kommandos zeigen, wie der Kernelcode von Version 4.3.5 auf 4.3.6 gepatcht wird. Dazu muss zuerst der Patch von Version 4.3 auf 4.3.5 rückgängig gemacht werden, bevor dann der Patch für Version 4.3.6 angewendet werden kann. Kernel-Patches beziehen sich üblicherweise nicht auf die letzte Major-Version, also hier auf die Version 4.3.

```
root# cd /usr/src/linux-4.3.5
root# bunzip2 -c patch-4.3.5.bz2 | patch -R -p1 --dry-run (Invers-Patch testen)
... keine Fehlermeldungen
root# bunzip2 -c patch-4.3.5.bz2 | patch -R -p1           (4.3.5 --> 4.3)
root# bunzip2 -c patch-4.3.6.bz2 | patch -p1 --dry-run    (Patch testen)
... keine Fehlermeldungen
root# bunzip2 -c patch-4.3.6.bz2 | patch -p1             (4.3 --> 4.3.6)
root# cd /usr/src
root# mv linux-4.3.5 linux-4.3.6
```

**pdf2ps** quelle.pdf [ziel.ps]

pdf2ps erzeugt aus einem PDF-Dokument eine PostScript-Datei. Wenn Sie auf die Angabe von ziel.ps verzichten, erhält die neue PDF-Datei den Namen quelle.pdf. Eine Alternative zu pdf2ps ist das Kommando pdftops aus der Poppler-Bibliothek. Um PostScript-Dateien in PDF-Dokumente umzuwandeln, verwenden Sie ps2pdf.

► -dLanguageLevel=n

gibt an, welcher PostScript-Level für die Ausgabe verwendet werden soll (standardmäßig PostScript Level 2, alternativ werden auch die Level 1 und 3 unterstützt).

**pdftk** datei1.pdf datei2.pdf ... kommando

pdftk manipuliert PDF-Dateien. Sie können damit Seiten extrahieren, mehrere PDF-Dokumente zusammenführen, PDF-Dokumente verschlüsseln bzw. die Verschlüsselung entfernen (wenn Sie das Passwort kennen) etc. Aus Platzgründen ist die Syntax hier vereinfacht dargestellt. Die komplette Syntax liefert man pdftk. Sie können mehrere Kommandos kombinieren, müssen dabei aber die Reihenfolge einhalten, in der die Kommandos hier beschrieben sind:

► input\_pw *password*

gibt das Passwort zu datei1.pdf an.

- ▶ `cat`  
fügt alle PDF-Dateien zusammen.
- ▶ `cat seitenliste`  
extrahiert die angegebenen Seiten. Mehrere Seitenbereiche werden durch Leerzeichen getrennt, also beispielsweise `1-5 7-10. end` bezeichnet die letzte Seite. Um die Seiten aus mehreren PDF-Dateien zusammenzufügen, geben Sie jeder Eingabedatei ein Kürzel und verweisen bei den Seitenangaben darauf (also beispielsweise `pdftk A=datei1.pdf B=datei2.pdf cat A2-4 B5 B7`). Wenn Sie nur Seiten mit geraden bzw. ungeraden Seitennummern wünschen, fügen Sie der Seitenangabe `even` oder `odd` hinzu, also etwa `1-10odd`.
- ▶ `background watermark.pdf`  
hinterlegt jede Ausgabeseite mit einer Seite von `watermark.pdf`. Wenn `watermark.pdf` weniger Seiten als die Ausgabe hat, wird die letzte Seite aus `watermark.pdf` wiederholt als Wasserzeichen verwendet. Die Ausgangs-PDF-Datei muss transparent sein, sonst ist das Wasserzeichen nicht sichtbar.
- ▶ `stamp stamp.pdf`  
schreibt über jede Ausgabeseite eine Seite aus `stamp.pdf`. Wenn `stamp.pdf` weniger Seiten als die Ausgabe hat, wird die letzte Seite aus `stamp.pdf` wiederholt. `stamp.pdf` muss transparent sein, sonst überdeckt sie die ursprüngliche PDF-Datei.
- ▶ `burst`  
erzeugt für jede Seite der PDF-Dateien eine eigene Datei mit dem Namen `page_n.pdf`, wobei `n` die Seitennummer ist.
- ▶ `output ergebnis.pdf`  
speichert die resultierende PDF-Datei unter dem Namen `ergebnis.pdf`.
- ▶ `owner_pw password` bzw. `user_pw password`  
verschlüsselt `ergebnis.pdf` mit dem angegebenen Passwort. Das `owner_pw`-Passwort gilt für den Ausdruck und andere PDF-Operationen, das `user_pw`-Passwort für das Öffnen der Datei.

## Beispiele

Das folgende Kommando liest die Seiten 10 bis 20 sowie 30 bis 40 aus `in.pdf` und schreibt sie in die neue Datei `out.pdf`:

```
user$ pdftk in.pdf cat 10-20 30-40 output out.pdf
```

Auch um mehrere PDF-Dateien aneinanderzufügen, verwenden Sie das Kommando `cat`:

```
user$ pdftk in1.pdf in2.pdf in3.pdf cat output out.pdf
```

Das folgende Beispiel erzeugt für jede einzelne Seite in `in.pdf` eine eigene PDF-Datei mit dem Namen `pg_n`, wobei `n` die Seitennummer ist:

```
user$ pdftk in.pdf burst
```

Das nächste Beispiel erzeugt eine verschlüsselte PDF-Datei. Die Datei kann zwar ohne das Passwort `xxx` gelesen, nicht aber ausgedruckt oder sonstwie bearbeitet werden. Wenn Sie selbst das Lesen der Datei schützen möchten, verwenden Sie statt `owner_pw` das Kommando `user_pw`.

```
user$ pdftk in.pdf output encrypted.pdf owner_pw xxx
```

Zuletzt wird `pdftk` dazu verwendet, eine PDF-Datei mit einem Wasserzeichen zu markieren:

```
user$ pdftk in.pdf background watermark.pdf output out.pdf
```

```
pdftops [optionen] quelle.pdf [ziel.ps]
```

`pdftops` aus der Poppler-Bibliothek, die unter Debian und Ubuntu im Paket `poppler-utils` bereitgestellt wird, erzeugt eine PostScript-Datei aus einem PDF-Dokument. Im Vergleich zu `pdf2ps` unterstützt es zahlreiche Optionen zur Beeinflussung der PostScript-Datei.

- ▶ `-eps`  
erzeugt eine EPS-Datei. Bei mehrseitigen PDF-Dokumenten muss mit `-f` und `-l` eine Seite ausgewählt werden.
- ▶ `-f n` und `-l n`  
gibt die erste und letzte Seite an (*first* und *last*).
- ▶ `-level n`  
gibt den gewünschten PostScript-Level an (1 bis 3).
- ▶ `-levelnsep`  
führt zusätzlich eine Farbseparierung durch. Dazu werden alle Farben in das CMYK-Format umgewandelt.



- ▶ `-opw passwort` bzw. `-upw passwort`  
gibt das Owner- bzw. User-Passwort an, um passwortgeschützte PDF-Dokumente zu verarbeiten.
- ▶ `-paper format`  
gibt das gewünschte Papierformat an (A3, A4, letter oder legal).
- ▶ `-paperw n` und `-paperh n`  
gibt die Papiergröße in Punkt an.

```
pdftotext [optionen] quelle.pdf [ziel.txt]
```

pdftotext extrahiert den Text aus einer PDF-Datei und speichert ihn in einer reinen Textdatei (standardmäßig `quelle.txt`). Die Formatierung sowie alle Bilder gehen dabei verloren. Auch pdftotext gehört zur Poppler-Bibliothek.

Die Optionen `-f`, `-l`, `-opw` und `-upw` haben dieselbe Bedeutung wie bei [pdftops](#) (siehe oben).

- ▶ `-layout`  
versucht, das Seitenlayout zu erhalten.
- ▶ `-npgbrk`  
verzichtet darauf, das Seitenende mit einem Sonderzeichen zu kennzeichnen.

```
pidof programmname
```

pidof ermittelt die Prozessnummern aller Instanzen eines namentlich genannten Prozesses.

- ▶ `-o %PPID`  
ignoriert den Elternprozess, bei Shell-Scripts also die Shell, in der das Script gerade ausgeführt wird.
- ▶ `-s`  
liefert nur den ersten passenden Prozess (*single shot*).

## Beispiel

Das folgende Kommando liefert die PIDs aller laufenden bash-Shells:

```
root# pidof /bin/bash
32329 21636 21600 3351 1739 922
```

**ping** [optionen] adresse  
**ping6** [optionen] adresse


ping sendet einmal pro Sekunde ein Netzwerkpaket (ECHO\_REQUEST gemäß dem Protokoll ICMP) an die angegebene Adresse. Wenn die Adresse erreichbar ist und Echo-Pakete nicht durch eine Firewall blockiert werden, empfängt ping Antwortpakete und gibt an, wie lange die Kommunikation hin und zurück gedauert hat. Standardmäßig läuft ping unbeschränkt, bis es durch **Strg**+**C** beendet wird. ping6 ist die IPv6-Variante zu ping.

- ▶ **-c n**  
sendet nur *n* Pakete und endet dann.
- ▶ **-i n**  
gibt die Intervallzeit zwischen zwei Paketen in Sekunden an. *n* ist eine Fließkommazahl, d. h., auch ping -i 0.1 ist möglich.
- ▶ **-n**  
zeigt nur die IP-Adresse, aber nicht den Hostnamen des Empfängers an.


### popd

Das bash-Kommando popd wechselt in ein zuvor mit pushd gespeichertes Verzeichnis zurück. Das Verzeichnis wird aus der Verzeichnisliste entfernt. popd und pushd werden üblicherweise nur in Shell-Scripts eingesetzt.

### powertop

Das interaktive Programm powertop aus dem gleichnamigen Paket analysiert, welche Prozesse am meisten Energie beanspruchen bzw. die CPU oder Festplatte am häufigsten aus einem Ruhemodus aufwecken. Während das Programm läuft, können Sie mit  zwischen mehreren Ergebnisseiten wechseln.

Die aus der Sicht der Energiesparfunktionen interessanteste Seite heißt TUNABLES. Dort zeigt powertop eine Liste von Einstellungen an, deren aktueller Zustand BAD bzw.

GOOD sein kann. Mit den Cursortasten können Sie nun einzelne Punkte auswählen und durch  umstellen. `powertop` zeigt dabei an, welches Kommando es ausführt.

Sie können nun schrittweise versuchen, einzelne Einstellungen zu ändern, und dann testen, welche Auswirkungen dies hat: Sinkt der mit einem Strommessgerät erfasste Energieverbrauch des zuvor aufgeladenen Notebooks tatsächlich spürbar? Verursachen die deaktivierten Funktionen Probleme? Lassen sich USB-Geräte weiterhin nutzen, funktioniert das Ein- und Ausschalten des WLAN-Adapters weiterhin, funktioniert das Audio-System ohne Störgeräusche etc.?

Die mit `powertop` durchgeführten Änderungen gelten nur bis zum nächsten Neustart des Rechners. Um die Energiesparmaßnahmen dauerhaft zu aktivieren, tragen Sie die von `powertop` angezeigten Kommandos (z. B. `echo '1' > /sys/xxx`) in eine Datei ein, die bei jedem Systemstart ausgeführt wird. Bei den meisten Distributionen eignet sich dazu `/etc/rc.d/rc.local`. Gegebenenfalls müssen Sie diese Datei erzeugen und mit `chmod a+x` als ausführbar kennzeichnen.

Eine radikale Lösung besteht darin, in `rc.local` anstelle einzelner Tuning-Kommandos `powertop --auto-tune` einzutragen. Dann führt `powertop` einfach alle bekannten Optimierungsmaßnahmen aus. Leider schießt `powertop` damit oft über das Ziel hinaus: Was nützt es, wenn das Notebook eine Stunde länger als bisher läuft, aber die Netzwerkverbindung nur noch unzuverlässig funktioniert?

Eine Liste aller möglichen Tuning-Kommandos erhalten Sie, wenn Sie `powertop` mit der Option `--html` ausführen. Das Kommando erzeugt dann nach einer Messzeit von ca. 20 Sekunden die HTML-Datei `powertop.html`, die neben diversen statistischen Daten auch eine Zusammenfassung aller Tuning-Parameter gibt.

```
ppa-purge ppa:ppaowner[/ppaname]
```

Das Ubuntu-spezifische Kommando `ppa-purge` aus dem gleichnamigen Kommando deaktiviert die angegebene private Paketquelle (PPA, siehe auch die Beschreibung des Kommandos `add-apt-repository`). Sofern die PPA alternative Versionen offizieller Pakete zur Verfügung stellt, werden diese Pakete entfernt und durch die Original-Pakete ersetzt.

## Beispiel

Das folgende Kommando deaktiviert die `libreoffice`-Paketquelle und ersetzt die von dieser Paketquelle stammenden Pakete durch Ubuntu-Standardpakete:

```
root# ppa-purge ppa:libreoffice/ppa
```

```
printenv [variable]
```

Gibt den Inhalt der angegebenen Umgebungsvariablen bzw. aller Umgebungsvariablen zeilenweise aus.

```
printf format para1 para2 para3 ...
```

printf erlaubt es, Ausgaben in der Syntax des C-Kommandos printf zu formatieren. Detaillierte Informationen zu den Formatierungsmöglichkeiten erhalten Sie mit `man 3 printf`.

### Beispiel

Das folgende Kommando gibt eine ganze Zahl, eine Fließkommazahl mit zwei Nachkommastellen sowie ein Zeilenende-Zeichen aus. Wenn Sie das Kommando auf einem System mit englischer Lokalisierung ausführen, müssen Sie die Fließkommazahl mit einem Dezimalpunkt anstelle des Kommas angeben.

```
root# printf "%d %.2f\n" 123 3,1415927
123 3,14
```

```
ps [optionen]
```

ps zeigt die Liste der laufenden Prozesse (Programme) und Kernel-Threads an. Das Kommando ist insbesondere im Zusammenspiel mit kill sehr praktisch, um hängende Programme gewaltsam zu beenden. ps ist mit zahllosen Optionen ausgestattet, die im Online-Manual (`man ps`) ausführlich beschrieben sind. Dort finden Sie auch Erklärungen dazu, was die zahlreichen Informationen bedeuten, die von ps ausgegeben werden. Eine Variante zu ps ist ps tree, das den Prozessbaum anzeigt und so auf einen Blick die Hierarchie der Prozesse klarmacht.

Im Gegensatz zu anderen Kommandos gibt es bei ps Optionen mit und ohne vorangehenden Bindestrich. Diese haben teilweise sogar eine unterschiedliche Bedeutung. (ps -a und ps a sind nicht gleichwertig!) Beide Optionstypen können in Gruppen gemischt werden, z. B. ps -A uL.

- ▶ a  
zeigt auch Prozesse anderer Benutzer an (nicht nur die eigenen).
- ▶ -A  
zeigt alle Prozesse an.

- ▶ `f`  
zeigt den Prozessbaum an.
- ▶ `-f -l` oder `l`  
zeigt diverse Zusatzinformationen an (Speicherbedarf, Priorität etc.).
- ▶ `-p n` oder `-p n1,n2,n3`  
zeigt nur Prozesse mit den angegebenen Prozessnummern (PIDs) an.
- ▶ `--ppid n`  
zeigt nur Prozesse an, die `n` als Parent-Prozess-ID haben, die also vom Prozess `n` gestartet wurden.
- ▶ `u`  
zeigt auch den Namen der Benutzer der jeweiligen Prozesse an. Die Option kann nicht mit `l` kombiniert werden.
- ▶ `x`  
zeigt auch Prozesse an, denen kein Terminal zugeordnet ist. Dabei handelt es sich unter anderem um Linux-interne Prozesse zur Verwaltung des Systems (engl. *Dæmons*).
- ▶ `Z`  
zeigt zusätzlich den SELinux-Kontext der Prozesse an.

## Beispiele

Die folgenden Beispiele zeigen elementare Anwendungen des `ps`-Kommandos:

```
root# ps ax           (zeigt alle Prozesse)
root# ps ax | grep ssh (zeigt Prozesse, in deren Beschreibung ssh vorkommt)
root# ps axu          (zeigt alle Prozesse inklusive Besitzer/Account)
```

Unter Linux hat der Kernel-Thread-Dämon `kthreadd` immer die Prozessnummer 2. Alle weiteren Kernel-Threads werden von `kthreadd` gestartet. Wenn Sie also nur die Kernel-Threads auflisten möchten, verwenden Sie `ps --ppid 2`:

```
root# ps --ppid 2      (zeigt nur Kernel-Threads)
```

**ps2pdf** [optionen] quelle.ps [ziel.pdf]

`ps2pdf` erzeugt aus einer beliebigen PostScript- oder EPS-Datei eine PDF-Datei. Wenn Sie auf die Angabe von `ziel.pdf` verzichten, bekommt die PDF-Datei denselben Namen wie die PostScript-Datei (aber die Kennung `.pdf`, also *quelle.pdf*).

► `-dEncodeColorImages=false`

verhindert jegliches Komprimieren von Bildern. Die resultierenden PDF-Dateien werden nun sehr groß. In Kombination mit der `/prepress`-Einstellung liefert `ps2pdf` nun PDFs in optimaler Druck- bzw. Belichtungsqualität.

► `-dPDFSETTINGS=/default | /screen | /printer | /prepress`

gibt an, ob das PDF-Dokument für eine beliebige Verwendung, für die Bildschirmdarstellung, für einen gewöhnlichen Ausdruck oder für eine Belichtung (Buchdruck etc.) optimiert werden soll. Der Preis für die höhere Qualität ist die zunehmende Dateigröße. Diese vier Voreinstellungen ersparen es Ihnen, zahlreiche Optionen einzeln einstellen zu müssen.

► `-In`

gibt die Auflösung für Bitmap-Fonts an (Einheit DPI, also *dots per inch*).

Unzählige weitere Optionen sind hier beschrieben:

<http://ghostscript.com/doc/current/Ps2pdf.htm>

### Beispiel

Das folgende Kommando erzeugt aus der PostScript-Datei `buch.ps` das PDF-Dokument `buch.pdf`:

```
user$ ps2pdf buch.ps
```

```
pssh [optionen] kommando
pscp [optionen] lokale_datei entfernte_datei
pnuke [optionen] muster
```

Das Python-Skript `pssh` aus dem gleichnamigen Paket führt ein Kommando parallel auf mehreren Servern aus. `pscp` kopiert Dateien von allen oder auf alle angegebenen Hosts. `pnuke` beendet auf allen Hosts mit `kill -9` ein Programm, dessen Name oder Muster angegeben wurde. Alle drei Kommandos setzen voraus, dass eine Authentifizierung mit Schlüsseln möglich ist oder dass sich das Hintergrundprogramm `ssh-agent` um die Authentifizierung kümmert.

► `-h hosts.txt`

gibt den Ort der Textdatei an, die zeilenweise die Hostnamen der Rechner enthält, auf denen das Kommando ausgeführt wird.

► `-o verzeichnis`

gibt ein Verzeichnis an, in dem `pssh` die Ausgaben der Kommandos bzw. `pscp` die zu kopierenden Dateien speichert. `pssh` und `pscp` richten für jeden Host eigene Ausgabedateien ein, aus deren Namen der Host hervorgeht.

### Beispiel

Das Beispiel geht davon aus, dass es sich bei allen Hosts um CentOS- oder RHEL-Systeme handelt. Auf diesen Rechnern wird nun mit `yum` ein Software-Update durchgeführt.

```
root# pssh -h hosts.txt yum update
```

**pstree** [optionen] [pid]

`pstree` aus dem Paket `psmisc` oder `pstree` gibt einen Baum mit allen Prozessen auf dem Bildschirm aus. Der Baum macht deutlich, welcher Prozess von welchem anderen Prozess gestartet wurde. Wenn eine Prozessnummer angegeben wird, beginnt der Baum an dieser Stelle, andernfalls beginnt er bei `init` oder `systemd`, also dem ersten Prozess, der beim Systemstart ausgeführt wird.

► `-h`

kennzeichnet den aktuellen Prozess und all seine Parent-Prozesse.

► `-p`

zeigt zu jedem Prozess auch die Prozessnummer (PID).

► `-u`

zeigt bei allen Prozessen, deren UID (User-ID) von der UID des Parent-Prozesses abweicht, den Benutzer- oder Account-Namen.

**pushd** verzeichnis

Das `bash`-Kommando `pushd` speichert das aktuelle Verzeichnis und wechselt anschließend in das angegebene Verzeichnis. Mit `popd` gelangen Sie in das ursprüngliche Verzeichnis zurück. `dirs` zeigt die Liste der gespeicherten Verzeichnisse an. `pushd` und `popd` werden überwiegend in Shell-Skripts eingesetzt.

```
pvccreate [optionen] device
```

Das LVM-Kommando `pvccreate` deklariert eine Partition bzw. ein Device als Physical Device (PV) zur späteren Nutzung in einer Volume Group (VG) (siehe auch `vgcreate` und `vgextend`).

`pvccreate` setzt voraus, dass die Partition bzw. das Device vorher als LVM-Partition gekennzeichnet wurde. In `fdisk` verwenden Sie dazu das Kommando `T` und den Code 8e. Bei `parted` lautet das erforderliche Kommando `set partitionsnummer lvm on`.

### Beispiel

Das folgende Kommando macht die Partition `/dev/sdc1` zu einem Physical Volume für LVM:

```
root# pvccreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
```

```
pvdisplay device
```

`pvdisplay` zeigt Detailinformationen zum angegebenen PV an.

```
pvrremove device
```

`pvrremove` entfernt die PV-Kennzeichnung eines ungenutzten PVs.

```
pvscan
```

`pvscan` listet alle PVs auf.

```
pwd
```

Das bash-Kommando `pwd` gibt das aktuelle Verzeichnis an.

```
pwgen [optionen] [n]
```

`pwgen` aus dem gleichnamigen Paket generiert eine ganze Liste zufälliger, aber leicht merkbarer Passwörter. Die Idee ist die, dass der Benutzer eines der Passwörter auswählt und dann den Bildschirm mit `[Strg]+[L]` löscht, bevor ihm jemand über die



Schulter blicken und das Passwort mitlesen kann. Im Vergleich zu den von `makepasswd` erzeugten Passwörtern sind die `pwgen`-Passwörter weniger zufällig und somit auch weniger sicher, aber für viele Anwendungsfälle sind sie immer noch ausreichend gut.

Die Länge der Passwörter beträgt standardmäßig acht Zeichen. Längere Passwörter generieren Sie, indem Sie den optionalen Parameter `n` angeben.

► -0

verwendet in den Passwörtern keine Ziffern.

► -1

liefert nur ein Passwort und nicht gleich 100.

► -B

vermeidet Buchstaben bzw. Ziffern, die je nach Schriftart leicht zu verwechseln sind, also O und 0 und I sowie l und 1. Das macht die Passwörter unsicherer.

► -y

baut in die Passwörter auch Sonderzeichen ein.

## Beispiel

Wie das Beispiel beweist, sind bereits die in den Standardeinstellungen erzeugten Passwörter keineswegs trivial:

```
user pwgen
Ohshu3yo Ea1wedoe OhCh2Zua Aili8ooc Xu4iiyix eug3Chee Gaesh2pu Eeth6mah
eMee0jio xiel6oob ob8uYah9 shaif0Ed uep8Eive lang3Eho thaiS7xa Sah3See4
...
```

Das folgende Kommando erzeugt ein 80 Zeichen langes Passwort, das auch Sonderzeichen enthält:

```
root# pwgen -1 -y 80
AhB9eiv]oo~s0go2ievaF7eePe6ooyoo7Cim%oh8eishiloosaeSeex+eeza9Iem7Ahchuhi6ahch[i1
```

**qalc** [ausdruck]

`qalc` aus dem gleichnamigen Paket berechnet den übergebenen mathematischen Ausdruck und liefert das Ergebnis zurück. Um die Verarbeitung von Sonderzeichen durch die `bash` zu vermeiden, müssen Sie die meisten Ausdrücke in Anführungszeichen stellen:

```
user$ qalc "(2+3)*4^5"
(2 + 3) * (4^5) = 5120
```

Wenn kein Ausdruck übergeben wird, aktiviert `qalc` einen interaktiven Modus. Es führt dann so lange Berechnungen aus, bis das Kommando mit `[Strg]+[D]` beendet wird.

```
user$ qalc
> 2^64
2^64 = approx. 1.8446744E19
> sin(0.2)
sin(0.2 * radian) = approx. 0.19866933
> sin(pi)
sin(pi * radian) = 0
> sqrt(2)
sqrt(2) = approx. 1.4142136
> 10 EUR to USD
10 * euro = $10.651
```

### qemu-img kommando

`qemu-img` hilft bei der Erzeugung und Verwaltung von Image-Dateien für virtuelle Festplatten. Als erster Parameter muss ein Befehl angegeben werden, z.B. `create` oder `convert`. Anschließend folgen je nach Befehl weitere Optionen, der Dateiname der Image-Datei etc.

- ▶ `convert [-f quellformat] -O zielformat quelldatei zieldatei`  
wandelt eine Image-Datei von einem Format in ein anderes um. Die ursprüngliche Datei bleibt dabei erhalten. Die Image-Datei darf aber während der Umwandlung nicht von einer virtuellen Maschine genutzt werden. Mit der zusätzlichen Option `-c` wird die Zieldatei komprimiert.
- ▶ `convert -f qcow2 -s snapshotname -O zielformat quelldatei zieldatei`  
überträgt nur den mit `-s` ausgewählten Snapshot eines QCOW2-Images in die neue Image-Datei.
- ▶ `create [-f raw/qcow2/qed] [-o opt1=val1,opt2=val2,...] datei größe`  
erzeugt eine neue Image-Datei im angegebenen Typ und in der gewünschten Größe. Die Größenangabe erfolgt standardmäßig in Byte. Optional können die Suffixe `k`, `M`, `G` oder `T` verwendet werden (für binäre `kByte`, `MByte`, `GByte` und `TByte`, das heißt, `M` entspricht  $1024 \times 1024$ ). Je nach Image-Format können diverse Zusatzoptionen angegeben werden. Die zur Auswahl stehenden Parameter ermitteln Sie mit `-o '?'`, wobei diese Option in einem vollständigen, syntaktisch korrekten Kommando angegeben werden muss:

```
qemu-img create -f qcow2 -o '?'test.img 1G
```

- ▶ `create -f qcow2 -o backing_file=basis.img overlay.img`  
erzeugt eine Overlay-Datei. Bei der Ausführung der virtuellen Maschine werden alle Änderungen in der Overlay-Datei gespeichert, die Basis-Datei bleibt unverändert.
- ▶ `info datei`  
liefert Informationen über die angegebene Image-Datei. Bei QCOW2-Images gibt info auch Auskunft über alle gespeicherten Snapshots.
- ▶ `resize datei +/-größe`  
vergrößert bzw. verkleinert die Image-Datei um die angegebene Größe. Das Kommando ist nur für RAW- und QCOW2-Images geeignet.
- ▶ `snapshot [optionen] snapshotname datei`  
bearbeitet Snapshots. Dieses Kommando ist nur für QCOW2-Images verfügbar. `-c` erzeugt einen neuen Snapshot, `-a` wendet einen Snapshot auf das Image an (wider ruft also alle Änderungen, die seit der Erstellung des Snapshots erfolgt sind), `-d` löscht den Snapshot, und `-l` listet alle Snapshots auf. Mit Ausnahme von `-l` dürfen Sie diese Kommandos allerdings nicht nutzen, während die Image-Datei von einer laufenden virtuellen Maschine genutzt wird!

## Beispiele

Das erste Kommando erzeugt eine QCOW2-Image-Datei. Ein großer Vorteil dieses Image-Formats besteht darin, dass die Datei anfänglich kaum Speicher auf dem Datenträger beansprucht und erst nach und nach wächst.

```
root# qemu-img create -f qcow2 disk.img 10G
```

Das zweite Kommando erzeugt aus einer QCOW2-Datei ein gleichwertiges Raw-Image:

```
user$ qemu-img convert -f qcow2 image.qcow2 -O raw image.raw
```

**qemu-nbd** [optionen] imagedatei

qemu-nbd bietet eine Image-Datei als Network Block Device an. kvm auf einem anderen Rechner kann dann über das Netzwerk auf das Image zugreifen, z. B. mit `kvm -hda nbd:nbd-hostname:1024`.

- ▶ `-b n.n.n.n`  
verbindet die Netzwerkschnittstelle mit der angegebenen IP-Adresse. Standardmäßig gilt die Adresse 0.0.0.0, d. h., der Netzwerkzugriff ist nicht auf das lokale Netzwerk limitiert.

- ▶ `-c device`  
verbindet die Image-Datei mit der angegebenen Device-Datei (*connect*).
- ▶ `-d device`  
löst die Verbindung zur angegebenen Device-Datei (*disconnect*).
- ▶ `-p n`  
verwendet den Port *n* (standardmäßig 1024).

### Beispiel

`qemu-nbd` wird hier dazu verwendet, um ein QCOW2-Image über eine NBD-Device-Datei anzubieten. Das vereinfacht die Verwendung des Images auf dem lokalen Rechner, ist aber für den Netzwerkeinsatz ungeeignet. Mit `parted` kann der virtuelle Datenträger nun wie eine lokale Festplatte bearbeitet werden. Das letzte Kommando gibt die Device-Datei wieder frei.

```
root# qemu-img create -f qcow2 disk.img 1G
root# modprobe nbd
root# qemu-nbd -c /dev/nbd0 disk.img &
root# parted /dev/nbd0
...
root# qemu-nbd -n /dev/nbd0
```

### `raspi-gpio` kommando

Das Kommando ermittelt bzw. verändert den Zustand der GPIO-Pins des Raspberry Pi. Es ist eine Alternative zum `gpio`-Kommando des WiringPi-Projekts. Das Kommando muss mit `root`-Rechten ausgeführt werden.

- ▶ `funcs [nr]`  
zeigt an, welche Funktionen der angegebene GPIO übernehmen kann. Wenn *nr* fehlt, liefert `raspi-gpio func` diese Daten für alle GPIOs.  
  
*nr* bezieht sich auf die Nummerierung der GPIOs gemäß der Dokumentation des Herstellers Broadcom (entspricht `gpio -h`). Die Nummern stimmen nicht mit den Pin-Nummern des J8-Headers überein!
- ▶ `get [nr]`  
zeigt den Zustand aller GPIOs bzw. des durch *nr* angegebenen GPIOs an.

- `help`  
zeigt einen relativ ausführlichen Hilfetext an. Eine `man`-Seite steht leider nicht zur Verfügung.

- `set nr zustand`  
aktiviert den gewünschten Zustand für den GPIO. Mögliche Werte sind:

`ip` = Input  
`op` = Output  
`dl` = Zustand 0 (drive low)  
`dh` = Zustand 1 (drive high)  
`a0` bis `a5` = alternative Funktion 0 bis 5  
`pu` = Pull-up-Widerstand aktiv  
`pd` = Pull-down-Widerstand aktiv  
`pn` = Pull-up/down-Widerstand deaktiviert (no pull)

Soweit sich sinnvolle Kombinationen ergeben, dürfen mehrere der obigen Schlüsselwörter auf einmal genannt werden, jeweils getrennt durch Leerzeichen. Welche alternativen Funktionen ein GPIO unterstützt und wie diese nummeriert sind, geht aus `raspi-gpio func` hervor.

## Beispiel

Das erste Kommando ermittelt, welche Funktionen der GPIO mit der Nummer 23 unterstützt. Auf dem Raspberry Pi ist dieser GPIO mit Pin 16 des J8-Headers verbunden. Das zweite Kommando nutzt diesen GPIO als simplen Signalausgang, wobei der Pegel auf High gesetzt wird (3,3 V). Eine über einen Vorwiderstand mit Pin 16 verbundene LED würde nun leuchten. Das dritte Kommando verifiziert den Zustand von GPIO 23.

```

root# raspigpio funcs 23
GPIO, DEFAULT PULL, ALT0, ALT1, ALT2, ALT3, ALT4, ALT5
23, DOWN, SDO_CMD, SD15, DPI_D19, SD1_CMD, ARM_RTCK, -
user$ raspigpio set 23 op dh
pi$ raspigpio get 23
GPIO 23: level=1 fsel=1 func=OUTPUT
  
```

**raspistill** [optionen]

Das unter Raspbian verfügbare Kommando `raspistill` nimmt ein Foto mit der Kamera auf, die als Erweiterung für den Raspberry Pi erhältlich ist.

- ▶ `-e jpg|bmp|gif|png`  
legt das gewünschte Bildformat fest (standardmäßig gilt jpg).
- ▶ `-h n`  
gibt die Bildhöhe an (standardmäßig 1944 Pixel).
- ▶ `-n`  
zeigt während der Aufnahme kein Vorschaufenster an.
- ▶ `-o datei`  
gibt den Namen der Bilddatei an.
- ▶ `-q n`  
gibt die JPEG-Bildqualität an (zwischen 0 = minimal und 100 = maximal).
- ▶ `-rot n`  
gibt die gewünschte Drehung des Bilds in Grad an (0 bis 359).
- ▶ `-t n`  
gibt an, nach welcher Zeitspanne in Millisekunden das Foto erstellt werden soll (standardmäßig nach 5000 ms, also nach 5 Sekunden). Um das Bild schnellstmöglich zu machen, geben Sie `-t 1` an. Die naheliegende Einstellung `-t 0` funktioniert nicht.
- ▶ `-tl n`  
gibt an, nach wie vielen Millisekunden jeweils weitere Fotos gemacht werden sollen.
- ▶ `-w n`  
gibt die Bildbreite an (standardmäßig 2592 Pixel).

Zur Steuerung der Bildqualität stehen unter anderem die folgenden Optionen zur Auswahl:

- ▶ `-br 0 bis 100`  
stellt die gewünschte Helligkeit an.
- ▶ `-co -100 bis 100`  
steuert den Kontrast.
- ▶ `-sa -100 bis 100`  
gibt ein Maß für die Farbsättigung an.

- `-sh -100 bis 100`

gibt die gewünschte Bildschärfe an.

### Beispiel

Das folgende Kommando erstellt bei minimaler Wartezeit und ohne die Anzeige eines Vorschaufensters ein Foto in einer Auflösung von  $800 \times 600$  Pixel:

```
pi$ raspistill -n -w 800 -h 600 -o foto.jpg
```

### **raspivid** [optionen]

Das unter Raspbian verfügbare Kommando `raspivid` nimmt ein Video im H264-Format mit der Kamera auf, die als Erweiterung für den Raspberry Pi erhältlich ist. Die Auflösung von Videos beträgt standardmäßig  $1920 \times 1080$  Pixel.

Viele Steuerungsoptionen stimmen mit dem oben beschriebenen Kommando `raspistill` überein, unter anderem `-br`, `-co`, `-h`, `-n`, `-o`, `-rot`, `-sa`, `-sh` und `-w`. An dieser Stelle konzentriere ich mich auf zwei `raspivid`-spezifische Optionen:

- `-fps n`

gibt an, wie viele Frames pro Sekunde aufgenommen werden sollen (standardmäßig 30, bei einer reduzierten Auflösung sind bis zu 90 Frames pro Sekunde möglich).

- `-t n`

gibt die gewünschte Länge des Videos in Millisekunden an (standardmäßig 5000, also 5 Sekunden).

### Beispiel

Das folgende Kommando nimmt ein Video in der Länge von 30 Sekunden bei einer Auflösung von  $1024 \times 768$  Pixeln auf:

```
pi$ raspivid -n -w 1024 -h 768 -t 30000 -o video.h264
```

### **rdiff-backup** [optionen] quellverzeichnis zielverzeichnis

`rdiff-backup` synchronisiert ähnlich wie `rsync` das Zielverzeichnis mit dem Quellverzeichnis. Im Unterschied zu `rsync` archiviert `rdiff-backup` bei wiederholter Ausführung auch alte Versionen von geänderten bzw. gelöschten Dateien, wobei aus

Platzgründen nur die Änderungen in komprimierter Form gespeichert werden, also in Form inkrementeller Backups. Das ermöglicht die Rekonstruktion von irrtümlich gelöschten oder überschriebenen Dateien.

Das Quell- und das Zielverzeichnis können sich auf Netzwerk-Servern befinden. In diesem Fall erfolgt die Kommunikation standardmäßig via `ssh`. Das Kommando `rdiff-backup` muss auch auf den externen Rechnern installiert sein. Bei der Angabe externer Verzeichnisse gilt nahezu dieselbe Syntax wie bei `rsync`. Der einzige Unterschied besteht darin, dass nach dem Hostnamen *zwei* Doppelpunkte angegeben werden müssen, also z. B. in der Form `user@backupserver::verzeichnis`.

► `-r zeit` bzw. `--restore-as-of zeit`

rekonstruiert die Daten, wie sie zum gegebenen Zeitpunkt waren. Die Wiederherstellung alter Dateien verursacht allerdings mit zunehmender Versionsanzahl einen erheblichen CPU-Aufwand und ist entsprechend langsam!

► `--remove-older-than zeit`

löscht inkrementelle Backup-Dateien, die älter als der angegebene Zeitpunkt sind. Den Zeitpunkt können Sie wahlweise absolut (z. B. 2013-12-31) oder relativ in Stunden (h), Tagen (D), Wochen (W) etc. angeben (siehe auch `man rdiff-backup` im Abschnitt `TIME FORMATS`). Statt eines konkreten Zeitpunkts können Sie mit `nB` auch angeben, wie viele Backup-Versionen maximal archiviert bleiben sollen. Standardmäßig löscht `rdiff-backup` immer nur *eine* Backup-Version. Wenn Sie mehrere Backup-Versionen auf einmal löschen möchten, müssen Sie zusätzlich die Option `--force` angeben.

## Beispiele

Das folgende Kommando erstellt ein Backup von `/home` im Verzeichnis `/home-backup`. Wenn dieses Kommando regelmäßig ausgeführt wird (z. B. täglich), werden automatisch auch alle Änderungen archiviert. Der Platzbedarf im Backup-Verzeichnis wächst damit entsprechend an.

```
root# rdiff-backup /home /home-backup
```

Das folgende Kommando rekonstruiert die gesicherten Daten in ein temporäres Verzeichnis:

```
root# rdiff-backup -r now /home-backup /tmp/home-aktuell
```

Das folgende Kommando stellt den Zustand des `/home`-Verzeichnisses so wieder her, wie er vor zehn Tagen war:

```
root# rdiff-backup -r 10D /home-backup/ /tmp/home-historisch
```



```
read [var1 var2 var3 ...]
```

`read` liest in `bash`-Scripts eine Zeile Text in die angegebenen Variablen. `read` erwartet die Daten aus der Standardeingabe. Wenn keine Variable angegeben wird, schreibt `read` die Eingabe in die Variable `REPLY`. Wenn genau eine Variable angegeben wird, schreibt `read` die gesamte Eingabe in diese eine Variable. Wenn mehrere Variablen angegeben werden, schreibt `read` das erste Wort in die erste Variable, das zweite Wort in die zweite Variable ... und den verbleibenden Rest der Eingabe in die letzte Variable. Wörter werden dabei durch Leer- oder Tabulatorzeichen getrennt.

Das `read`-Kommando sieht keine Möglichkeit vor, einen Infotext als Eingabeaufforderung auszugeben. Deswegen ist es zweckmäßig, den Anwender vor der Ausführung von `read`-Kommandos mit `echo -n` über den Zweck der Eingabe zu informieren.

```
readonly
```

Das `bash`-Kommando `readonly` zeigt die schreibgeschützten Variablen der Shell an. Variablen können mit `declare -r` vor Veränderungen geschützt werden.

```
reboot [optionen]
```

`reboot` beendet alle laufenden Prozesse und startet den Rechner anschließend neu. `reboot` entspricht `shutdown -r now`.

```
recode Zeichensatz1..Zeichensatz2 datei  
recode Zeichensatz1..Zeichensatz2 < quelle > ziel
```

`recode` führt eine Zeichensatzkonvertierung von Zeichensatz 1 nach Zeichensatz 2 durch. `recode -l` liefert eine umfangreiche Liste aller unterstützten Zeichensätze.

## Beispiele

Das folgende Kommando konvertiert die DOS-Datei `dosdat` in eine Linux-Datei mit dem Latin-1-Zeichensatz:

```
user$ recode ibmpc..latin1 < dosdat > linuxdat
```

Das zweite `recode`-Beispiel ersetzt in der Datei `windowsdat` alle Zeilenenden (CR plus LF, also *Carriage Return* und *Line Feed*) durch das unter Linux übliche Zeilenende (nur LF). Der eigentliche Zeichensatz wird nicht geändert. Die resultierende Datei wird in `linuxdat` gespeichert.

```
user$ recode latin1/cr-lf..latin1 < windowsdat > linuxdat
```

recode liest die im Zeichensatz Latin-1 codierte Textdatei `latin1dat` und speichert sie als UTF-8-Datei (Unicode):

```
user$ recode latin1..u8 < latin1dat > utf8dat
```

### rename

Bei den meisten Distributionen ist ein `rename`-Kommando vorinstalliert, das bei der Umbenennung mehrerer Dateien hilft. Allerdings kommen je nach Distribution unterschiedliche Varianten mit unterschiedlicher Syntax zur Anwendung. Im Folgenden beschreibe ich die beiden gängigsten Varianten, zuerst das unter Debian, Raspbian und Ubuntu übliche Perl-Script aus dem Paket `rename` und dann das unter CentOS, Fedora, RHEL und SUSE gängige Kommando `rename` aus dem Paket `util-linux`.

#### rename unter Debian, Raspbian und Ubuntu

Die Syntax dieser `rename`-Variante sieht so aus:

```
rename [optionen] rename_expression dateien
```

Dabei gibt die `rename_expression` in einer `sed`-ähnlichen Syntax an, wie mit den in den weiteren Parametern genannten Dateien verfahren werden soll. Die Details der Syntax können Sie mit `man perlexpr` nachlesen. Ich beschränke mich an dieser Stelle auf zwei Beispiele:

► `-f` bzw. `-force`

überschreibt vorhandene Dateien.

► `-n`

zeigt an, welche Dateien umbenannt würden, führt aber keine Änderungen durch.

Das folgende Beispiel benennt alle `*.jpg`-Dateien in `*.jpeg`-Dateien um:

```
user$ rename -n s/\.jpg/\.jpeg/ *.jpg
```

Das zweite Kommando ersetzt alle Großbuchstaben durch Kleinbuchstaben:

```
user$ rename -n y/A-Z/a-z/ *
```

#### rename unter CentOS, Fedora, RHEL und SUSE

Die `rename`-Syntax unter CentOS, Fedora, RHEL und SUSE ist simpler:

```
rename [optionen] find replace dateien
```

Das Kommando ersetzt einfach den Ausdruck `find` im Dateinamen durch `replace`. Es gibt keine Möglichkeit, die Wirkung von `rename` vorweg auszuprobieren.

► `-s`

verändert auch die Namen von Dateien, auf die symbolische Links zeigen.

► `-v`

zeigt die durchgeführten Namensänderungen an.

Um wieder `.jpg` in `.jpeg`-Dateien umzubenennen, führen Sie `rename` so aus:

```
user$ rename jpg jpeg *.jpg
```

**renice** *n* *pid*

`renice` verändert die Priorität des Prozesses mit der angegebenen PID-Nummer. *n* ist wahlweise die neue Priorität (ein Wert zwischen `-20` und `20`) oder ein Deltawert mit positivem oder negativem Vorzeichen (z. B. `+3` oder `-2`). Nur `root` darf die Priorität von Prozessen erhöhen. Die höchste Priorität ist `-20`, die kleinste ist `20`.

**repoquery** [*optionen*]

Das Red-Hat-spezifische Kommando `repoquery` aus dem Paket `yum-utils` ermittelt Metadaten zu Paketen aus YUM-Paketquellen. Es spielt keine Rolle, ob das Paket bereits installiert ist oder nicht.

► `-i paketname`

ermittelt Informationen über das angegebene Paket. Das Ergebnis sieht so ähnlich aus wie jenes von `rpm -qi`, liefert aber im Feld `Repository` zusätzlich auch die Paketquelle, aus der das Paket stammt.

► `--location paketname`

liefert eine HTTP-Adresse, von der das Paket heruntergeladen werden kann.

## Beispiel

Auf dem Testrechner stammt das Paket `joe` aus der EPEL-Paketquelle für RHEL-Systeme:

```
root# repoquery -i joe
Name       : joe
Version    : 3.7
URL        : http://sourceforge.net/projects/joe-editor/
Repository : epel
...
```

Unter Fedora ist `repoquery` kein eigenständiges Kommando mehr, sondern ein Subkommando von `dnf`:

```
root# dnf repoquery joe
Name       : joe
Version    : 4.0
...
```

### reset

`reset` stellt die Schrift im Textterminal wieder her, wenn diese durch die Ausgabe von Sonderzeichen zerstört worden ist. `reset` setzt auch die Terminaleinstellungen auf die früher mit `setterm -store` gespeicherte Grundeinstellung zurück.

### resize2fs [optionen] device [größe]

`resize2fs` verändert die Größe eines ext2/ext3/ext4-Dateisystems. Für die Größenangabe sind die Schreibweisen *nK*, *nM* und *nG* für kByte, MByte und GByte zulässig. Wenn Sie keine Größe angeben, passt `resize2fs` das Dateisystem an die Größe der zugrunde liegenden Partition bzw. des Logical Volumes an (ideal für Vergrößerungen).

Vergrößerungen des Dateisystems sind im laufenden Betrieb möglich. Für eine Verkleinerung müssen Sie das Dateisystem aushängen und vorher `fsck` ausführen. Beachten Sie, dass Sie bei einer Vergrößerung *vorher* die zugrunde liegende Partition oder das LV vergrößern müssen, bei einer Verkleinerung die Partition oder das LV aber erst *nachher* verkleinern dürfen!

### Beispiel

Im folgenden Beispiel wird zuerst ein Logical Volume mit `lvextend` vergrößert. Anschließend wird das darin enthaltene Dateisystem mit `resize2fs` vergrößert:

```
root# lvextend -L 40G /dev/mapper/vg1-test
Extending logical volume test to 40,00 GB
Logical volume test successfully resized
root# resize2fs /dev/mapper/vg1-test
Das Dateisystem auf /dev/mapper/vg1-test ist auf /test eingehängt;
Online-Größenveränderung nötig
```

Führe eine Online-Größenänderung von /dev/mapper/vg1-test auf 10485760 (4k) Blöcke durch.  
Das Dateisystem auf /dev/mapper/vg1-test ist nun 10485760 Blöcke groß.

### restorecon [optionen] dateien

restorecon stellt den für ein Verzeichnis vorgesehenen SELinux-Kontext wieder her. Das ist erforderlich, wenn SELinux vorübergehend deaktiviert war oder nachdem Sie Dateien so verschoben oder kopiert haben (z. B. mit `cp -a`), dass keine automatische Einstellung des SELinux-Kontexts möglich war. Um den SELinux-Kontext verzeichnisunabhängig festzulegen, verwenden Sie das Kommando [chcon](#).

- ▶ `-0`  
erwartet, dass die Dateinamen durch O-Bytes und nicht durch Tabulator- oder Leerzeichen getrennt sind. Das hilft in Kombination mit `find -print0` bei der Verarbeitung von Dateien, deren Namen Leerzeichen enthalten.
- ▶ `-e verzeichnis`  
überspringt dieses Verzeichnis. Die Option kann wiederholt werden, um mehrere Verzeichnisse von den Kontext-Änderungen auszunehmen.
- ▶ `-r` bzw. `-R`  
berücksichtigt auch alle Unterverzeichnisse (*recursive*).
- ▶ `-v`  
zeigt die durchgeführten Änderungen an (*verbose*).

### Beispiel

Das folgende Kommando stellt den korrekten SELinux-Kontext für alle Dateien im DocumentRoot-Verzeichnis des Webserver Apache korrekt ein:

```
root# restorecon -R -v /var/www/html/*
```

### rftkill kommando

rftkill aus dem gleichnamigen Paket ermöglicht es, Bluetooth-, WLAN- und Mobilfunkadapter ein- und auszuschalten, beispielsweise um bei einem Notebook Strom für gerade nicht benötigte Funktionen zu sparen.

► `block index/typ`

deaktiviert das durch die Index-Nummer spezifizierte Gerät bzw. alle Geräte eines bestimmten Typs. Zulässige Typen sind `all`, `wifi` = `wlan`, `bluetooth`, `uwb` = `ultrawide-band`, `wimax`, `wwan`, `gps` und `fm`.

► `event`

wartet auf `rftkill`-Ereignisse und gibt diese aus.

► `list`

listet alle Funk-Adapter und deren Zustand auf. `soft blocked` bedeutet, dass das Gerät durch `rftkill` ein- und ausgeschaltet werden kann. `hard blocked` bedeutet, dass `rftkill` das Gerät nicht steuern kann, z. B. weil es mit einem mechanischen Schalter verbunden ist.

► `unblock index/typ`

aktiviert das bzw. die angegebenen Geräte, die wie bei `block` angegeben werden.

**Beispiel**

Auf dem Testrechner stehen zwei Bluetooth- und ein WLAN-Adapter zur Verfügung. Einer der Bluetooth-Adapter wird aktiviert.

```
root# rftkill list
0: tpacpi_bluetooth_sw: Bluetooth
   Soft blocked: yes
   Hard blocked: yes
1: hci0: Bluetooth
   Soft blocked: yes
   Hard blocked: no
2: phy0: Wireless LAN
   Soft blocked: yes
   Hard blocked: no
root# rftkill unblock 1
```

**rm** [optionen] dateien

`rm` löscht die angegebenen Dateien. Verzeichnisse werden – sofern nicht die Option `-r` verwendet wird – nicht gelöscht. Zum Löschen einzelner Verzeichnisse ist das Kommando `rmdir` vorgesehen. Wenn Dateien mit Sonderzeichen gelöscht werden sollen, müssen die Dateinamen in einfache Apostrophe eingeschlossen werden. Die wichtigsten Optionen vom `rm` sind:

- ▶ `-f`  
löscht ohne Rückfragen (auch Verzeichnisse). Vorsicht!
- ▶ `-i` bzw. `--interactive` oder `-v` bzw. `--verbose`  
zeigt vor dem Löschen jeder einzelnen Datei eine Rückfrage an.
- ▶ `-r` bzw. `-R` bzw. `--recursive`  
löscht auch Dateien in allen Unterverzeichnissen (Vorsicht!). Wenn dabei der gesamte Inhalt des Unterverzeichnisses gelöscht wird, wird auch das Unterverzeichnis selbst eliminiert.

### Beispiele

Das folgende Kommando löscht alle Backup-Dateien (Dateien, die mit `~` enden) im aktuellen Verzeichnis:

```
user$ rm *~
```

`rm` löscht die Datei oder das Verzeichnis `backup`. Wenn `backup` ein Verzeichnis ist, werden auch alle darin enthaltenen Unterverzeichnisse und Dateien gelöscht!

```
user$ rm -r backup
```

`rm` löscht alle Dateien, die mit dem Doppelkreuz `#` beginnen. Die Apostrophe sind erforderlich, damit die Shell `#` nicht als Kommentar interpretiert.

```
user$ rm '#'*
```

**rmmdir** [optionen] verzeichnis

`rmmdir` löscht das angegebene Verzeichnis. `rmmdir` kann nur ausgeführt werden, wenn das Verzeichnis leer ist. Eventuell vorhandene Dateien müssen vorher mit `rm` gelöscht werden. Die wichtigste Option lautet:

- ▶ `-p` bzw. `--parents`  
löscht auch Unterverzeichnisse im angegebenen Verzeichnis, sofern die Verzeichnisse – mit Ausnahme von Unterverzeichnissen – leer sind.

**rmmod** [optionen] modulname

`rmmod` entfernt das angegebene Modul aus dem Kernel. Das funktioniert nur, wenn das Modul nicht mehr benötigt wird.

```
route [optionen]
route add/del [-net/-host] zieladresse [optionen]
```

route zeigt die Routing-Tabelle für das Netzwerkprotokoll IP an, fügt einen neuen Eintrag hinzu (add) bzw. entfernt einen Eintrag (del). Die Zieladresse wird wahlweise in numerischer Form oder als Netzwerkname angegeben. Als Zieladresse ist auch das Schlüsselwort default erlaubt, um eine Standardregel zu definieren (z. B. für das Gateway).

Das Paket route gilt als veraltet. Verwenden Sie stattdessen nach Möglichkeit das Kommando `ip route`!

- ▶ `-host`  
adressiert einen einzelnen Rechner.
- ▶ `-n`  
zeigt nur numerische IP-Adressen an und kontaktiert keinen Nameserver zur Auflösung der zugeordneten Namen.
- ▶ `-net`  
adressiert ein Netzwerk.
- ▶ `dev schnittstelle`  
gibt die gewünschte Netzwerkschnittstelle an (z. B. eth0). Wenn dev die letzte Option ist, kann auf das Schlüsselwort verzichtet werden (also `route add -net 192.34.34.34 netmask 255.255.255.0 eth1`).
- ▶ `gw routingadresse`  
leitet die Pakete an den angegebenen Router weiter.
- ▶ `netmask maske`  
gibt die gewünschte Netzwerkmaske an (z. B. 255.255.255.0).

## Beispiel

Das erste route-Kommando definiert die IP-Adresse 192.168.0.1 als Standard-Gateway. Das zweite Kommando zeigt die neue Routing-Tabelle an.

```
root# route add default gw 192.168.0.1
root# route -n
Kernel IP routing table
Destination Gateway    Genmask         Flags Metric Ref Use Iface
192.168.0.0 0.0.0.0      255.255.255.0   U      0      0    0 eth0
0.0.0.0     192.168.0.1 0.0.0.0         UG     0      0    0 eth0
```



**rpcinfo** [optionen]

rpcinfo führt einen Remote Procedure Call (RPC) durch und liefert das vom RPC-Server zurückgegebene Ergebnis. rpcinfo wird unter anderem dazu eingesetzt, um die NFS-Server-Konfiguration zu überprüfen.

► **-p** [hostname]

ermittelt eine Liste aller aktiven NFS- und RPC-Dienste auf dem lokalen Rechner bzw. auf einem externen Rechner.

**rpm** optionen [dateiname/paketname]

rpm installiert, aktualisiert oder entfernt RPM-Pakete bzw. zeigt Informationen zu diesen Paketen an. rpm steht auf allen Distributionen zur Verfügung, die auf RPM-Paketen basieren, z. B. Fedora, Red Hat und SUSE.

► **-e** bzw. **--erase**

entfernt ein vorhandenes Paket. Wenn Sie zusätzlich die Option **--nodeps** verwenden, wird das Paket selbst dann entfernt, wenn andere Pakete davon abhängig sind.

► **-i** bzw. **--install**

installiert die angegebene(n) Paketdatei(en). Der Installationsort kann durch die zusätzliche Option **--root** *verzeichnis* verändert werden. Wenn es sich beim angegebenen Paket um ein Quellpaket handelt, werden der Programmcode und die Konfigurationsdateien in */usr/src* installiert. Der Installationsort kann in diesem Fall durch eine Veränderung von */etc/rpmrc* eingestellt werden.

Mit der Zusatzoption **--test** führt rpm keine Veränderungen durch, sondern gibt nur aus, was bei einer tatsächlichen Installation passieren würde.

rpm versucht, automatisch die Integrität signierter Pakete zu überprüfen. Dazu muss aber der öffentliche Schlüssel der Signatur zur Verfügung stehen (siehe die Option **--import**). Wenn rpm auf die Kontrolle verzichten soll, übergeben Sie die Option **--nosignature**.

Mit der Zusatzoption **--nodeps** können Sie ein Paket selbst dann installieren, wenn rpm glaubt, dass diverse Abhängigkeiten nicht erfüllt sind. (Falls zwei Pakete gegenseitig voneinander abhängig sind, besteht keine Notwendigkeit, **--nodeps** einzusetzen. Installieren Sie einfach beide Pakete gleichzeitig!)

Die Zusatzoption `--ignorearch` ermöglicht eine Installation auch dann, wenn `rpm` erkennt, dass die CPU-Architektur nicht stimmt. Das ist bisweilen erforderlich, um 32-Bit-Pakete auf einer 64-Bit-Distribution zu installieren.

Die Zusatzoption `--force` erzwingt eine Installation in jedem Fall, selbst dann, wenn `rpm` glaubt, dass das Paket bereits installiert ist.

Mit der Zusatzoption `--noscripts` verzichtet `rpm` auf die automatische Ausführung von Installationsprogrammen. Unter Umständen funktioniert das gerade installierte Programm dann aber nicht ordnungsgemäß.

► `--import`

installiert den angegebenen GPG-Schlüssel. `rpm` kann anhand dieses Schlüssels die Integrität signierter Pakete überprüfen. Die erforderlichen Schlüsseldateien finden Sie auf allen Websites, die signierte RPM-Pakete zur Verfügung stellen.

► `-q` bzw. `--query`

liefert Informationen über installierte Pakete bzw. über den Inhalt einer Paketdatei. In der einfachsten Form gibt `rpm -q paketname` den exakten Paketnamen inklusive Versionsnummer des angegebenen Pakets aus.

Mit `rpm -qf datei` stellen Sie fest, zu welchem Paket eine bestimmte Datei Ihres Systems gehört. Das ist besonders bei Konfigurationsdateien sehr nützlich. Beispielsweise liefert `rpm -qf /sbin/hwclock` den Paketnamen `util-linux-n`.

`rpm -qp paket.rpm` liefert Informationen über ein noch *nicht* installiertes Paket. Als Parameter muss in diesem Fall der Dateiname einer `*.rpm`-Datei verwendet werden.

`rpm -q --whatprovides attribut` bzw. `rpm -q --whatrequires attribut` sucht nach allen installierten Paketen, die das angegebene Abhängigkeitsattribut zur Verfügung stellen bzw. erfordern. Diese zwei Optionen sind nicht geeignet, um noch nicht installierte Pakete zu durchsuchen!

`rpm -q --provides paket` bzw. `rpm -q --requires` erstellt eine Liste aller Attribute, die das angegebene Paket zur Verfügung stellt bzw. benötigt. Die Optionen können auch für noch nicht installierte Paketdateien (zusätzliche Option `-p`) eingesetzt werden.

Bei allen Abfragevarianten, die als Ergebnis ein Paket liefern, steuern Sie durch Zusatzoptionen die Informationsfülle zum gefundenen Paket:

- i: Kurzbeschreibung des Pakets, komprimierte Größe
- l: Liste aller Dateien des Pakets
- lv: wie oben, aber mit Dateigröße, Zugriffsrechten etc.

- c: Liste aller Konfigurationsdateien des Pakets
- d: Liste aller Dateien mit Online-Dokumentation zum Paket
- scripts: Liste aller Installationsprogramme

Wenn Sie einfach nur eine unsortierte Liste aller installierten Pakete benötigen, führen Sie `rpm -qa` ohne weitere Parameter aus. `rpm -qa --last` sortiert alle Pakete nach dem Installationsdatum (das zuletzt installierte Paket zuerst).

► -U bzw. --upgrade

aktualisiert ein bereits vorhandenes Paket. Von den veränderten Konfigurationsdateien des bisherigen Pakets werden automatisch Backup-Dateien erzeugt; alle anderen Dateien des bisherigen Pakets werden durch die neue Version ersetzt bzw. gelöscht. Mit der zusätzlichen Option `--oldpackage` können Sie ein neueres Paket durch ein älteres ersetzen. Auch beim Upgrade erreichen Sie durch `--nodeps`, dass `rpm` Paketabhängigkeiten ignoriert.

► -V bzw. --verify

überprüft, ob sich irgendwelche Dateien eines Pakets seit der Installation verändert haben. Das Kommando liefert eine Liste aller veränderten Dateien. Dokumentationsdateien werden nicht überprüft.

## Beispiele

Das folgende Kommando installiert das angegebene Paket:

```
root# rpm -i abc-2.0.7-1.i686.rpm
```

Das zweite `rpm`-Beispiel liefert eine Liste aller installierten Pakete, wobei die Pakete nach dem Installationsdatum sortiert werden (das zuletzt installierte Paket steht zuoberst in der Liste):

```
root# rpm -qa --last
```

`rpm -q --whatprovides` verrät, welches bereits installierte Paket das Attribut `mysql.so` zur Verfügung stellt:

```
root# rpm -q --whatprovides mysql.so
php-mysql
```

Wenn Sie nicht wissen, zu welchem Paket die Datei `/etc/magic` gehört, ermitteln Sie diese Informationen mit `rpm -qf`. Anschließend liefert `rpm -qi` eine Beschreibung des Pakets:

```
root# rpm -qf /etc/magic
file-n.n
```

```
root# rpm -qi file
```

```
...
```

The file command is used to identify a particular file according to the type of data contained by the file.

```
rpm2archive name.rpm
```

```
rpm2cpio name.rpm
```

`rpm2archive` erzeugt aus einem als Datei vorliegenden RPM-Paket ein komprimiertes TAR-Archiv (`name.rpm.tgz`) mit allen Dateien des Pakets. `rpm2archive` ist praktisch, um einzelne Dateien aus einem Paket zu extrahieren, ohne das ganze Paket zu installieren.

`rpm2archive` steht erst ab der RPM-Version 4.12 bzw. ab Fedora 23 zur Verfügung. Bei älteren Distributionen müssen Sie sich mit dem äquivalenten Kommando `rpm2cpio` behelfen. Es erzeugt eine `cpio`-Datei, deren Inhalt Sie mit `cpio` extrahieren können.

```
rsync [optionen] quelle
```

```
rsync [optionen] quelle ziel
```

`rsync` kopiert Dateien vom Quell- in das Zielverzeichnis bzw. synchronisiert die beiden Verzeichnisse. Bei der ersten Syntaxvariante (kein Zielverzeichnis) werden die Dateien in das lokale Verzeichnis kopiert. `rsync` überträgt die Dateien wahlweise lokal oder verschlüsselt über das Netzwerk. Die Netzwerkvariante setzt voraus, dass auf dem zweiten Rechner ein SSH- oder RSYNC-Server läuft. Bevor das Kopierkommando startet, muss das Passwort angegeben werden. Für die Angabe der Quell- und Zielverzeichnisse bestehen die folgenden Syntaxvarianten:

<code>datei1 datei2</code>	lokale Dateien
<code>verzeichnis</code>	lokales Verzeichnis
<code>host:verz</code>	Verzeichnis auf dem Rechner <code>host</code>
<code>user@host:verz</code>	wie oben, aber SSH-Login unter dem Namen <code>user</code>
<code>rsync://user@host/verz</code>	Kommunikation mit RSYNC-Server
<code>rsync://user@host:port/verz</code>	RSYNC-Server am angegebenen Port

► `-a` bzw. `--archive`

kopiert den Verzeichnisinhalt rekursiv und erhält alle Dateiinformationen. Die Option ist eine Kurzfassung für `-rlptgoD`.

► `--delete`

löscht im Zielverzeichnis Dateien bzw. Unterverzeichnisse, die im Quellverzeichnis nicht mehr existieren (Vorsicht!).

- ▶ `-D` bzw. `--devices`  
kopiert auch Zeichen- und Block-Device-Informationen.
- ▶ `-e kommando` bzw. `--rsh=kommando`  
gibt an, welches externe Shell-Programm eingesetzt wird, um `rsync` auf dem Zielrechner auszuführen. In der Regel wird dies `ssh` sein. Wenn Sie an das Kommando zusätzliche Optionen übergeben möchten, setzen Sie es in Hochkommas, z. B. `-e "ssh -p 1234"`.
- ▶ `--exclude=muster`  
gibt ein Muster für Dateien an, die nicht kopiert werden sollen. `--exclude="*.o"` schließt beispielsweise `*.o`-Dateien aus. (Zur Formulierung komplexerer Muster siehe man `rsync`.)
- ▶ `-g` bzw. `--group`  
gibt den Dateien im Zielverzeichnis dieselbe Gruppenzugehörigkeit wie bei Quelldateien.
- ▶ `-l` bzw. `--links`  
dupliziert symbolische Links des Quellverzeichnisses auch im Zielverzeichnis.
- ▶ `-o` bzw. `--owner`  
gibt den Dateien im Zielverzeichnis denselben Besitzer, den auch die Dateien im Quellverzeichnis haben.
- ▶ `-p` bzw. `--perms`  
gibt den Dateien im Zielverzeichnis dieselben Zugriffsrechte wie bei Quelldateien.
- ▶ `-r` bzw. `--recursive`  
kopiert rekursiv auch alle Unterverzeichnisse.
- ▶ `-t` bzw. `--times`  
gibt den Dateien im Zielverzeichnis dieselbe Änderungszeit wie bei den Quelldateien. Auf diese Weise kann `rsync` bei einem neuerlichen Aufruf eindeutig erkennen, welche Dateien sich geändert haben.
- ▶ `-u` bzw. `--update`  
ignoriert Dateien, die im Zielverzeichnis bereits existieren und neuer sind als die Quelldatei.
- ▶ `-v` bzw. `--verbose`  
liefert ausführliche Informationen darüber, was gerade passiert.

## ► -W bzw. --whole-file

kopiert bei Änderungen die gesamte Datei. Bei schnellen Netzwerkverbindungen kann das die Synchronisationsgeschwindigkeit ein wenig erhöhen.

Standardmäßig überträgt `rsync` nur die Änderungen und minimiert so die Menge der übertragenen Daten. Die Option `-W` vergrößert die zu übertragende Datenmenge, reduziert aber den Aufwand zur Feststellung der Änderungen. `-W` gilt automatisch, wenn es sich bei den Quell- und Zielverzeichnissen um lokale Verzeichnisse handelt. In diesem Fall erreichen Sie mit `--no-whole-file`, dass `rsync` dennoch den inkrementellen Änderungsmodus nutzt.

## Beispiele

Das folgende Kommando synchronisiert alle Dateien aus dem Verzeichnis `mydata` mit dem gleichnamigen Backup-Verzeichnis. Dateien, die in `mydata` gelöscht wurden, werden auch im Backup-Verzeichnis gelöscht.

```
user$ rsync -av --delete mydata/* /backup/mydata/
```

Das zweite Beispiel synchronisiert das lokale Verzeichnis `verz1` mit dem Verzeichnis `verz2` auf dem Rechner `mars.sol`. Für die Passwordeingabe ist `ssh` verantwortlich. (Es muss also das Login-Passwort des Benutzers `username` auf dem Rechner `mars.sol` eingegeben werden.)

```
user@saturn.sol$ rsync -e ssh -az verz1/ mars.sol:verz2/
user@mars.sol's password: *****
```

**scp** [optionen] quelle ziel

`scp` kopiert eine Datei zwischen zwei Rechnern in einem Netzwerk. Die Übertragung erfolgt verschlüsselt. `scp` setzt voraus, dass auf dem zweiten Rechner ein SSH-Server läuft. Die Quell- und Zieldatei bzw. das Zielverzeichnis werden wie folgt angegeben:

<code>datei</code>	lokale Datei
<code>verzeichnis</code>	lokales Zielverzeichnis
<code>host:</code>	Zielrechner (kopiert Dateien in das Heimatverzeichnis)
<code>host:datei</code>	Datei auf dem angegebenen Rechner
<code>host:verzeichnis</code>	Zielverzeichnis auf dem angegebenen Rechner
<code>user@host:datei</code>	Datei des Nutzers <code>user</code> auf dem angegebenen Rechner
<code>user@host:verzeichnis</code>	Zielverzeichnis des Nutzers <code>user@host</code>

Anstelle eines Hostnamens kann auch die IP-Adresse angegeben werden. IPv6-Adressen müssen in eckige Klammern gesetzt werden.

- ▶ `-4` bzw. `-6`  
erzwingt das IPv4- bzw. das IPv6-Protokoll, falls beide Varianten zur Auswahl stehen.
- ▶ `-l limit`  
beschränkt die Übertragungsgeschwindigkeit (in kByte/s).

### Beispiel

Das folgende Kommando kopiert die lokale Datei `abc.txt` in das Verzeichnis `~/efg` des Rechners `mars`:

```
user@uranus$ scp abc.txt mars:efg/
user@mars's password: *****
```

**sdptool** [optionen] kommando

Mit dem `sdptool` aus dem `bluez`-Paket können Sie Anfragen gemäß dem Service Discovery Protocol (SDP) an Bluetooth-Geräte senden. Am häufigsten kommt dabei das Kommando `browse` zum Einsatz.

- ▶ `browse bt-mac`  
liefert Informationen über das durch seine MAC-Adresse angegebene externe Bluetooth-Gerät. Die MAC-Adresse ermitteln Sie mit `hcitool scan`.
- ▶ `browse local`  
liefert detaillierte Informationen über den lokalen Bluetooth-Adapter.

### Beispiel

`sdptool` stellt fest, dass es sich beim angegebenen Bluetooth-Gerät um ein Eingabegerät (Human Interface Device) handelt, das die Protokolle L2CAP und HIDP unterstützt, also das *Logical Link Control and Adaptation Protocol* und das *Human Interface Device Protocol*.

```
user$ sdptool browse 60:FB:42:FC:BB:8C
Service Name: Wireless Mouse
Service Class ID List:
  "Human Interface Device" (0x1124)
Protocol Descriptor List:
  "L2CAP" (0x0100)
    PSM: 17
  "HIDP" (0x0011)
...
```

**sealert** [optionen]

sealert hilft bei der Diagnose von SELinux-Regelverstößen. Das Kommando setzt voraus, dass der Hintergrunddienst `setroubleshootd` läuft. Dieser Dienst protokolliert SELinux-Probleme in `/var/log/messages`, beispielsweise so:

```
root# less /var/log/messages
...
Aug 31 12:45:45 fedora setroubleshoot: SELinux is preventing /usr/sbin/httpd
    from read access on the file tst. For complete SELinux messages run
    sealert -l dccb472d-6dd8-49d2-b7d7-2658e082c805
```

Nun können Sie mit `sealert` die Ursache des Problems ergründen:

```
root# sealert -l dccb472d-6dd8-49d2-b7d7-2658e082c805
SELinux is preventing /usr/sbin/httpd from read access on the file index.html.

**** Plugin catchall_boolean (89.3 confidence) suggests ****
If sie möchten allow httpd to read user content Then you must tell SELinux
about this by enabling the 'httpd_read_user_content' boolean. You can read
'user_selinux' man page for more details. Do setsebool -P
httpd_read_user_content 1
...
```

Leider ist das Ergebnis sprachlich wie inhaltlich sehr mangelhaft und selten eine echte Hilfe.

► **-a logdatei**

analysiert die angegebene Loggingdatei und liefert Informationen zu allen darin protokollierten SELinux-Ereignissen.

► **-l id**

gibt an, zu welchem SELinux-Ereignis `sealert` Informationen liefern soll. `-l '*'` behandelt alle protokollierten SELinux-Regelverstöße.

**sed** [optionen] kommando [< quelle > ziel]

`sed` ist ein sogenannter Stream-Editor. Das Kommando wird normalerweise als Textfilter verwendet, um bestimmte Zeichen oder Zeichenkombinationen im Ausgangstext zu finden und zu bearbeiten (löschen, durch andere Zeichen ersetzen etc.). Die Steuerung des `sed` erfolgt durch Kommandos, die entweder auf alle Zeilen des Textes angewendet werden oder nur auf solche Zeilen, die bestimmten Bedingungen entsprechen.



Der Umgang mit `sed` ist leider nicht ganz einfach. Selbst eine Beschreibung der elementaren Syntaxoptionen würde den Rahmen dieser Referenz bei Weitem sprengen. Deswegen beschränke ich mich hier auf zwei Beispiele. Eine ausgezeichnete englischsprachige Einführung in den Umgang mit `sed` finden Sie hier:

<http://www.grymoire.com/Unix/Sed.html>

### Beispiel

Im ersten Beispiel löscht `sed` die erste bis dritte Zeile der Datei `test` und leitet den Rest der Datei an die Standardausgabe. Dabei steht `d` für das `sed`-Kommando *delete*. `1,3` gibt den Adressbereich an, in dem dieses Kommando wirkt.

```
user$ sed 1,3d < test
```

Im zweiten Beispiel ersetzt `sed` alle »a« durch »A«. Dabei steht `s` für das Kommando *regular find and replace*. Die in `/` eingeschlossenen Texte sind das Suchmuster und der einzusetzende Text. Da vor dem Kommando keine Adressangabe steht, wird es auf alle Zeilen angewandt.

```
user$ sed s/a/A/ < test
```

Wie Sie `sed` zum Kopieren bzw. Umbenennen von Dateien einsetzen können, zeigt ein weiteres Beispiel bei der Beschreibung des `cp`-Kommandos.

```
seq [optionen] ende
seq [optionen] start ende
seq [optionen] start delta ende
```

`seq` liefert eine Sequenz von Ziffern. Die Funktionsweise geht aus den folgenden Beispielen hervor. Beachten Sie, dass bei Fließkommazahlen der Endwert aufgrund von Rundungsfehlern unter Umständen gerade nicht erreicht wird, wie dies im letzten Beispiel demonstriert ist.

#### ► -s zeichenkette

gibt an, durch welches Zeichen die Ergebnisse getrennt werden. Standardmäßig geschieht das durch `\n`, also durch einen Zeilenumbruch.

### Beispiele

Die folgenden Beispiele zeigen die Anwendung von `seq`. Dabei kommt aus Platzgründen als Trennzeichen ein Leerzeichen zum Einsatz. Normalerweise (also ohne die Option `-s`) wird jede Zahl in einer eigenen Zeile ausgegeben.

```
user$ seq -s ' ' 5
1 2 3 4 5
user$ seq -s ' ' 4 9
4 5 6 7 8 9
user$ seq -s ' ' 10 5 30
10 15 20 25 30
user$ seq -s ' ' 1 0.1 2
1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
```

Zur Formulierung von `for`-Schleifen in `bash`-Scripts sollten Sie nach Möglichkeit die Schreibweisen `{start..ende}` oder `{start..ende..delta}` nutzen, also beispielsweise:

```
user$ for i in 100..150..10; do echo $i; done
100 110 120 130 140 150
```

### **service** dienst kommando

`service` erlaubt es, unter Debian, Fedora, openSUSE, RHEL und Ubuntu Dienste zu starten und zu stoppen, die vom `Init`-System verwaltet werden. Zu den erlaubten Kommandos zählen `start`, `stop`, `restart`, `reload` und `status`. `service --status-all` liefert eine alphabetische Liste mit dem Status aller Dienste. `service` ist zu `Init-V`, `Systemd` und `Upstart` kompatibel.

### Beispiele

Das folgende Kommando startet unter Fedora den Webserver Apache neu:

```
root# service httpd restart
```

Unter Debian und Ubuntu ist für Apache das `Upstart`-Script `apache2` zuständig:

```
root# service apache2 restart
```

### **sestatus** [optionen]

`sestatus` zeigt den Zustand des SELinux-Systems an. Wenn das Kommando ohne weitere Optionen verwendet wird, liefert es eine Zusammenfassung des aktuellen Zustands.

#### ► -b

zeigt zusätzlich den Zustand aller booleschen Parameter der SELinux-Regeln an. Diese Parameter können mit `setsebool` verändert werden.

## ► -v

zeigt zusätzlich Kontextinformationen zu den in `/etc/sestatus.conf` aufgelisteten Dateien und Prozessen an.

### Beispiel

Auf dem Testrechner ist SELinux aktiv, die Einhaltung der Regeln wird sichergestellt (Current mode = enforcing). Es kommt das Regelset targeted zum Einsatz.

```
root# sestatus -b
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Max kernel policy version:    29

Policy booleans:
abrt_anon_write                off
abrt_handle_event              off
abrt_upload_watch_anon_write   on
antivirus_can_scan_system      off
...
```

### set

Das bash-Kommando `set` zeigt alle der Shell bekannten Variablen an (inklusive der Umgebungsvariablen, die auch mit `printenv` angezeigt werden können). `set -x` bewirkt, dass die bash vor der Ausführung jedes Kommandos anzeigt, wie die interne Kommandozeile nach der Berücksichtigung von alias-Abkürzungen und der Expansion der Dateinamen aussieht.

### setcap [optionen] capability dateiname

`setcap` aus dem Paket `libcap-ng-utils` legt bei ausführbaren Dateien fest, welche Operationen für das Programm zulässig sind (Capabilities). Wenn statt `capability` die Option `-r` übergeben wird, werden alle früher festgelegten Capabilities gelöscht. Capabilities setzt ein Dateisystem mit Extended Attributes voraus. Bei ext-Dateisystemen muss also die `mount`-Option `user_xattr` verwendet werden.

### Beispiel

Das Netzwerkkommando `ping` ist bei manchen Distributionen (z. B. in Ubuntu bis zumindest Version 15.10) mit dem `setuid`-Bit ausgestattet, sodass es von gewöhnlichen Benutzern verwendet werden kann. Sobald Sie dieses Bit löschen, kann nur noch `root` mit `ping` arbeiten:

```
user$ sudo chmod u-s /bin/ping
user$ ping yahoo.de
ping: icmp open socket: Die Operation ist nicht erlaubt
```

Anstatt nun das unsichere `setuid`-Bit wieder zu setzen, reicht es auch, dem Kommando `ping` mit `setcap` den Zugriff auf Netzwerkfunktionen des Kernels zu geben. Mit `getcap` können Sie nachsehen, welche Capabilities ein Kommando hat.

```
user$ sudo setcap cap_net_raw=ep /bin/ping
user$ getcap /bin/ping
/bin/ping = cap_net_raw+ep
user$ ping yahoo.de
PING yahoo.de (212.82.102.24) 56(84) bytes of data.
...
rtt min/avg/max/mdev = 58.054/58.054/58.054/0.000 ms
```

Bei vielen aktuellen Distributionen ist `ping` mittlerweile standardmäßig wie oben beschrieben mit Capabilities ausgestattet, z. B. unter Debian, Fedora und openSUSE.

**setenforce** 0|1|Enforcing|Permissive

`setenforce` schaltet SELinux zwischen den Modi `Enforcing` (1) und `Permissive` (0) um. Im `Permissive`-Modus werden Regelübertretungen zwar protokolliert, das betroffene Programm kann aber ungehindert weiterarbeiten. Im `Enforcing`-Modus hindert SELinux das Programm hingegen, durch SELinux-Regeln verbotene Vorgänge auszuführen.

Wenn Sie SELinux vollständig deaktivieren möchten, müssen Sie in `/etc/selinux/config` die Einstellung `SELINUX=disabled` vornehmen und den Rechner neu starten. Beachten Sie, dass eine spätere Reaktivierung schwierig ist, weil Sie dann den SELinux-Kontext aller neuen Dateien richtigstellen müssen. Deswegen ist der `Permissive`-Modus in der Regel vorzuziehen.

**setfacl** [optionen] [aclaktion] dateiname

`setfacl` verändert die erweiterten Zugriffsrechte der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem ACLs (*Access Control Lists*)

unterstützt. Bei ext3/ext4-Dateisystemen muss dazu die `mount`-Option `acl` verwendet werden.

Das Kommando wird üblicherweise zur Durchführung einer der vier folgenden Aktionen eingesetzt:

- ▶ `-m aclregel`  
fügt den vorhandenen ACLs eine weitere Regel (eine *Access Control Entity*, kurz ACE) hinzu. (`m` steht hier für *modify*.)
- ▶ `-M aclregeldatei`  
wie `-m`, liest die Regel aber aus einer Datei. Regeldateien müssen wie die Ausgabe von `getfacl` aussehen.
- ▶ `-x aclregel`  
löscht die angegebene ACL-Regel.
- ▶ `-X aclregeldatei`  
wie `-x`, liest die Regel aber aus einer Datei.

Der vereinfachte Aufbau einer ACL-Regel (einer ACE) geht aus der folgenden Tabelle hervor. Die vollständige Syntax ist in `man setfacl` dokumentiert.

<code>[u:]uid</code>	<code>[:rechte]</code>	ändert Rechte für einen Benutzer.
<code>g:gid</code>	<code>[:rechte]</code>	ändert Rechte für eine Gruppe.
<code>o[:]</code>	<code>[:rechte]</code>	ändert die Rechte für alle anderen Benutzer.
<code>m[:]</code>	<code>[:rechte]</code>	stellt die ACL-Maske ( <i>Effective Right Mask</i> ) ein.

Die rechte bestehen aus bis zu drei Buchstaben: `r` für *read*, `w` für *write* und `x` für *execute*. Um einem Benutzer oder einer Gruppe alle Rechte zu entziehen, geben Sie `-` an.

Wenn Sie der gesamten Regel `d:` voranstellen, gilt sie für die Standard-ACLs. Statt der Buchstaben `u`, `g`, `o` und `m` können Sie auch die Schlüsselwörter `user`, `group`, `other` und `mask` verwenden.

`setfacl` kann durch weitere Optionen gesteuert werden:

- ▶ `-b` bzw. `--remove-all`  
entfernt alle ACL-Regeln.
- ▶ `-d`  
wendet die übergebene Regel auf die Standard-ACL an.
- ▶ `-k` bzw. `--remove-default`  
entfernt die Standard-ACL-Regeln.

► -n

verzichtet auf die sonst bei jeder ACL-Regeländerung automatische Neuberechnung der ACL-Maske.

► --restore=*datei*

wendet die in der Datei angegebenen ACL-Regeln auf die Dateien im aktuellen Verzeichnis an. Eine ACL-Backup-Datei erstellen Sie mit `getfacl -R`.

► -R

wendet die angegebene Regel rekursiv auf alle Dateien und Unterverzeichnisse an.

### Beispiele

Bei einem Dateisystem mit ACLs gelten normalerweise die Standardzugriffsrechte, die oft auch als *minimale ACL* bezeichnet werden. `getfacl` zeigt diese Rechte in ACL-Form an:

```
user$ touch datei1
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
group::r--
other::r--
user$ ls -l datei1
-rw-r--r-- 1 kofler kofler ... datei2
```

Mit `setfacl` definieren Sie nun zusätzliche Zugriffsregeln. Die folgenden Kommandos geben der Benutzerin gabi sowie allen Mitgliedern der Gruppe docuteam Schreib- und Lesezugriff auf die Datei, verbieten aber der Benutzerin kathrin jeglichen Zugriff:

```
user$ setfacl -m gabi:rw datei1
user$ setfacl -m g:docuteam:rw datei1
user$ setfacl -m kathrin:- datei1
```

Die Rechtestliste von `getfacl` ist nun schon etwas länger. Bei `ls -l` folgt den üblichen Zugriffsbuchstaben das Zeichen +, um darauf hinzuweisen, dass es ACL-Regeln gibt.

```
user$ getfacl datei1
# file: datei1
# owner: kofler
# group: kofler
user::rw-
user:gabi:rw-
user:kathrin:---
group::r--
```

```
group:docuteam:rw-
mask::rw-
other::r--
user$ ls -l datei1
-rw-rw-r--+ 1 kofler kofler ... datei1
```

```
setfattr [optionen] dateiname
```

setfattr ändert die erweiterten Attribute der angegebenen Dateien bzw. Verzeichnisse. Das funktioniert nur, wenn das Dateisystem EAs (*Extended Attributes*) unterstützt. Bei ext3/ext4-Dateisystemen muss dazu die mount-Option `user_xattr` verwendet werden.

- ▶ `-n attributname` bzw. `--name=attributname`  
gibt den Namen des zu verändernden Attributs an. Dem eigentlichen Namen muss `user.` vorangestellt werden (also `-n user.meinattribut`).
- ▶ `-v wert` bzw. `--value=wert`  
gibt den Wert an, der im Attribut gespeichert werden soll.
- ▶ `-x attributname`  
löscht das angegebene Attribut.
- ▶ `--restore=datei`  
wendet die in der Datei angegebenen EA-Definitionen auf die Dateien im aktuellen Verzeichnis an. Eine EA-Backup-Datei erstellen Sie mit `getfattr -R -d`.

## Beispiele

Die folgenden Beispiele zeigen, wie Sie mit setfattr Attribute speichern und diese mit getfattr auslesen. Die Anzahl der Attribute pro Datei ist in ext3-Dateisystemen beschränkt.

```
user$ touch datei2
user$ setfattr -n user.language -v de datei2
user$ setfattr --name=user.charset --value=utf8 datei2
user$ getfattr -d datei2
# file: datei2
user.charset="utf8"
user.language="de"
```

getfattr liefert normalerweise nur Attribute, deren Name mit »user.« beginnt. Wenn Sie andere Attribute sehen möchten, müssen Sie deren Namen durch `-n` oder deren Muster durch `-m` angeben:

```
user$ getfattr -n security.selinux -d tst
# file: tst
security.selinux="user_u:object_r:user_home_t:s0^000"
```

### **setfont** fontfile

setfont liest bei einigen Distributionen eine Schriftartdatei und aktiviert diesen Font im VGA-Textmodus. setfont kann nur in einer Textkonsole und nicht unter X verwendet werden. Brauchbare Schriften sind default8x16 (für den 25-zeiligen Textmodus) und default8x9 (für den 50-zeiligen Textmodus).

### **setsebool** [optionen] bool1=wert1 bool2=wert2 ...

setsebool verändert boolesche Parameter zur Laufzeit. Mit booleschen Parametern lässt sich das Verhalten einiger SELinux-Regeln verändern. Welche boolesche Parameter es gibt, können Sie mit sestatus -b herausfinden.

#### ► -N

speichert die Veränderung, aktiviert sie aber nicht. Die Änderung wird somit erst mit dem nächsten Neustart des Rechners wirksam.

#### ► -P

führt die Veränderung durch und speichert die Einstellung bleibend. Damit gilt die Änderung sofort und auch in Zukunft, also nach dem nächsten Neustart.

### Beispiel

Das folgende Kommando erlaubt dem Programm Squid, als transparenter Proxy zu agieren:

```
root# setsebool -P squid_use_tproxy 1
```

### **setterm** [option]

setterm verändert diverse Einstellungen des Terminals. Wenn das Kommando ohne die Angabe einer Option ausgeführt wird, zeigt es eine Liste aller möglichen Optionen an. Nützliche Optionen zur Shell-Programmierung sind:



- ▶ `--background farbe`  
stellt die Hintergrundfarbe ein. Dabei sind die Farben `black`, `red`, `green`, `yellow`, `blue`, `magenta`, `cyan` und `white` erlaubt. Damit die Einstellung auch nach Farbausgaben durch `ls` oder `grep` gültig bleibt, muss die Option mit `--store` kombiniert werden.
- ▶ `--blank n`  
aktiviert nach *n* Minuten ohne Eingabe den Bildschirmschoner (nur für Textkonsolen).
- ▶ `--bold on|off`  
aktiviert bzw. deaktiviert die fette Schrift. In Textkonsolen erscheint der Text zwar nicht fett, aber immerhin in einer anderen Farbe als der sonstige Text.
- ▶ `--clear`  
löscht den Inhalt des Terminals.
- ▶ `--default`  
stellt Farben und Textattribute auf die Standardeinstellung zurück.
- ▶ `--foreground farbe`  
stellt die Textfarbe ein.
- ▶ `--half-bright on|off`  
stellt hervorgehobene Schrift ein/aus.
- ▶ `--inversescreen on|off`  
invertiert die Darstellung in der Textkonsole (schwarzer Text auf weißem Hintergrund) bzw. stellt wieder den Normalzustand her.
- ▶ `--reverse on|off`  
stellt inverse Schrift ein/aus.
- ▶ `--store`  
merkt sich die Farbeinstellungen (Optionen `--foreground` und `--background`), so dass diese auch erhalten bleiben, wenn Kommandos wie `ls` oder `grep` vorübergehend andere die Farben verwenden. `--store` ist nur in den Linux-Textkonsolen gültig, nicht aber innerhalb von Terminalfenstern. Auch dort werden die Einstellungen nicht bleibend in einer Datei gespeichert, sondern gehen mit dem Logout verloren.

- ▶ `--underline on|off`

stellt unterstrichene Schrift ein/aus.

### Beispiel

Wenn die folgende Zeile in `.bashrc` eingefügt wird, verwenden Textkonsolen nach dem Login einen weißen Hintergrund und schwarze Schrift:

```
# in .bashrc einfügen
setterm -background white -foreground black -store
```

### **sfdisk** [optionen]

`sfdisk` listet die Partitionen einer Festplatte auf bzw. partitioniert die Festplatten neu. Das Kommando eignet sich besonders zur automatisierten Partitionierung (z. B. in einem Script) bzw. dazu, eine vorhandene Partitionierung auf eine zweite Festplatte zu übernehmen, etwa beim Einrichten eines RAID-Verbunds.

`sfdisk` kommt nur mit MBR-Partitionstabellen zurecht und ist für Datenträger mit einer GUID-Partitionstabelle (GPT) ungeeignet!

- ▶ `-d [device]`

listet alle Partitionen aller Festplatten bzw. der angegebenen Festplatte auf. Das Ausgabeformat eignet sich zur Weiterverarbeitung durch `sfdisk`.

- ▶ `--force`

führt die Partitionierung auch dann durch, wenn `sfdisk` glaubt, dass die Partition bzw. Festplatte verwendet wird, bzw. wenn Partitionsgrenzen nicht mit Zylinder-grenzen übereinstimmen.

- ▶ `-l [device]`

listet alle Partitionen aller Festplatten bzw. der angegebenen Festplatte in einer gut lesbaren Form auf. Mit der zusätzlichen Option `-uS`, `-uB`, `-uC` oder `-uM` verwendet `sfdisk` Sektoren (512 Byte), Blöcke (1024 Byte), Zylinder oder MByte als Maßeinheit.

- ▶ `-s device`

liefert die Anzahl der Blöcke (zu je 1024 Byte) der angegebenen Festplatte oder Partition.

- ▶ `-uS|-uB|-uC|-uM`

verwendet Sektoren (512 Byte), Blöcke (1024 Byte), Zylinder oder MByte als Maßeinheit.

► *device < partitionierungstabelle*

partitioniert die Festplatte anhand der übergebenen Tabelle neu. Die Partitionierungstabelle muss dasselbe Format wie die Ausgabe von `sfdisk -d` haben.

### Beispiel

Das folgende Kommando formatiert die zweite Festplatte (`/dev/sdb`) exakt genauso wie die erste Festplatte (`/dev/sda`). Das setzt voraus, dass die zweite Festplatte zumindest genauso groß wie die erste ist. Alle Partitionen bzw. Daten auf der zweiten Festplatte gehen durch dieses Kommando verloren.

```
root# sfdisk -d /dev/sdb | sfdisk /dev/sda
```

```
sftp [optionen] sshserver
sftp [optionen] user@sshserver:dateiname
```

sftp ist eine Alternative zum unsicheren `ftp`-Kommando. In der ersten Syntaxvariante stellt sftp eine Verbindung zum angegebenen SSH-Server her. Nach dem Login können Sie interaktiv Dateien zwischen dem lokalen Rechner und dem SSH-Server übertragen. In der zweiten Syntaxvariante überträgt sftp die angegebene Datei nach der Passwortheingabe direkt auf den lokalen Rechner.

► *-b batchdatei*

führt die in der Batch-Datei angegebenen Kommandos aus.

```
sgdisk [option] device
```

sgdisk hilft bei der Bearbeitung von GUID-Partitionstabellen auf Festplatten und SSDs. Anders als `gdisk` und `parted` wird das Kommando ausschließlich durch Optionen gesteuert und ist nicht für den interaktiven Einsatz gedacht. Vielmehr können Sie mit sgdisk Partitionstabellen Script-gesteuert erstellen oder manipulieren. Trotz des ähnlich klingenden Namens erwartet sgdisk ganz andere Optionen als `sfdisk`. sgdisk arbeitet mit binären MByte und GByte, also MiB und GiB in der IEC-Schreibweise.

► *-a n*

gibt an, zu welchem Vielfachen von Sektoren neue Partitionen ausgerichtet werden. Standardmäßig verwendet sgdisk den Wert 2048. Das entspricht einem MByte.

► *-b datei*

schreibt ein Backup der Partitionstabelle in eine binäre Datei.

► `-d nr`

löscht die angegebene Partition.

► `-G`

weist allen Partitionen neue, zufällige GUIDs zu. Das ist nach dem Klonen einer Festplatte erforderlich.

► `-l datei`

partitioniert den Datenträger entsprechend der in der Datei gespeicherten Partitionstabelle. Die Datei muss vorher mit der Option `-b` erzeugt werden. Alle bisher auf dem Datenträger vorhandenen Partitionen werden gelöscht. Die Kombination der Kommandos `sgdisk -b` und `sgdisk -l` kann dazu verwendet werden, um das Partitionslayout einer Festplatte auf einen zweiten gleich großen Datenträger zu übertragen, z. B. beim Einrichten eines RAID-Systems.

► `-n nr:start:end`

legt die neue Partition mit der Nummer `nr` an, die von der angegebenen Start- bis zur Endposition reicht. Standardmäßig wird `start` und `end` in Sektoren zu 512 Byte angegeben. Alternativ sind auch Positionsangaben in binären MByte und GByte möglich. `sgdisk -n 3:10G:35G` erzeugt eine neue Partition, die 25 GByte groß ist.

Den Start- und Endpositionen können die Zeichen `+` und `-` vorangestellt werden. `+` ermöglicht Angaben, die relativ zum Default-Startsektor sind. `-` ermöglicht Angaben, die vom Ende der Festplatte gerechnet werden. `sgdisk -n 4:+1M:-20G` erzeugt eine Partition, die 1 MByte hinter der letzten Partition beginnt und 20 GByte am Ende des Datenträgers frei lässt.

## Beispiel

Die folgenden Kommandos erstellen zuerst ein Backup der aktuellen Partitionstabelle auf dem Datenträger `/dev/sdb` und richten dann die neue Partition 3 ein:

```
root# sgdisk -l partbackup /dev/sdb
root# sgdisk -n 3:70G:430G /dev/sdb
```

## shasum dateien

`sha1sum`, `sha224sum`, `sha256sum`, `sha384sum` und `sha512sum` berechnen Prüfsummen zu allen angegebenen Dateien. Die Zahl im Kommando gibt die Bitlänge der Prüfsumme an (160 Bits bei `sha1sum`). Die Prüfsummen werden verwendet, um nach der Übertragung einer Datei sicherzustellen, dass die Datei unverändert ist.

## Beispiel

Das folgende Kommando berechnet eine 256-Bit-Prüfsumme für eine ISO-Datei:

```
root# sha256sum image.iso
6ad62fe91eefa4315f852eb1bc8732bc341d95b47f6b4d1f5e362d3629fc981b  image.iso
```

## shift [n]

Das bash-Kommando `shift` schiebt die dem Shell-Script übergebene Parameterliste durch die vordefinierten Variablen `$1` bis `$9`. Wenn `shift` ohne Parameter verwendet wird, werden die Parameter um eine Position verschoben, andernfalls um *n* Positionen.

`shift` wird nur zur Script-Programmierung eingesetzt. Das Kommando ist eine wertvolle Hilfe, wenn mehr als neun Parameter angesprochen werden sollen. Einmal mit `shift` aus den Variablen geschobene Parameter können nicht mehr angesprochen werden. Sie werden auch aus der Variablen `$*` entfernt.

## showmount

Ohne weitere Optionen liefert das Kommando `showmount`, wenn es auf einem NFS-Server ausgeführt wird, eine Liste aller Clients, die die freigegebenen Verzeichnisse per NFS 3 nutzen. Beachten Sie, dass NFS-4-Verzeichnisse nicht in das Ergebnis einfließen und nur bei der Variante `showmount -e` berücksichtigt werden.

### ► -a

liefert zu jedem NFS-3-Client auch die IP-Adresse.

### ► -e [hostname]

liefert eine Liste aller Verzeichnisse, die per NFS genutzt werden können, unabhängig davon, ob dieser Dienst momentan genutzt wird. Wenn Sie einen Hostnamen angeben, wird die Liste der Verzeichnisse ermittelt, die auf diesem Rechner verfügbar sind. `showmount -e` funktioniert auch für NFS 4.

## Beispiel

```
root# showmount -e
/nfsexport      *.lan,10.0.0.0/24
/nfsexport/fotos *.lan,10.0.0.0/24
/nfsexport/iso   *
```

```
shutdown [optionen] zeitpunkt [nachricht]
```

`shutdown` fährt den Rechner herunter. Als Zeitpunkt muss entweder eine Uhrzeit (hh:mm), die Anzahl der Minuten, gerechnet von der aktuellen Zeit (+m), oder das Schlüsselwort `now` (also sofort) angegeben werden. `shutdown` kann nur von `root` ausgeführt werden. Oft ist Linux so konfiguriert, dass Anwender ohne `root`-Rechte den Rechner mit `[Alt]+[Strg]+[Entf]` neu starten können.

`shutdown` informiert alle anderen Benutzer darüber, dass das System in Kürze abgeschaltet wird, und lässt keine neuen Logins mehr zu. Anschließend werden alle Prozesse gewarnt, dass sie in Kürze gestoppt werden. Einige Programme (etwa `emacs`, `vi` ...) nutzen diese Warnung und speichern alle offenen Dateien in Backup-Kopien.

- ▶ `-c`  
versucht, einen bereits eingeleiteten `shutdown`-Vorgang abubrechen. Das ist dann möglich, wenn die Zeit bis zum Beginn des `shutdown`-Prozesses noch nicht abgelaufen ist.
- ▶ `-f`  
wie `-r`, aber schneller.
- ▶ `-F`  
bewirkt eine Überprüfung der Dateisysteme beim nächsten Neustart. `shutdown` erzeugt zu diesem Zweck die Datei `/forcefsck`. Ist sie vorhanden, wird bei den meisten Distributionen beim Neustart eine Dateisystemüberprüfung auslöst.
- ▶ `-h`  
hält das System nach dem Herunterfahren an. Der Rechner reagiert dann nicht mehr auf Eingaben. Auf dem Bildschirm erscheint die Meldung *system halted*. Anschließend wird der Rechner durch ein ACPI-Signal ausgeschaltet.
- ▶ `-n`  
führt den Shutdown unter Umgehung des Init-Systems besonders schnell aus.
- ▶ `-r`  
veranlasst nach dem Herunterfahren des Systems einen Neustart.
- ▶ `-t sekunden`  
bestimmt, wie lange zwischen der Warnnachricht und dem Kill-Signal für die Prozesse gewartet werden soll (normalerweise 20 Sekunden).

**sleep** zeit

sleep versetzt das laufende Programm für die angegebene Zeit in den Ruhezustand. Das Programm konsumiert in dieser Zeit praktisch keine Rechenzeit. Die Zeitangabe erfolgt normalerweise in Sekunden. Optional können die Buchstaben m, h oder d angehängt werden, um die Zeit in Minuten, Stunden oder Tagen anzugeben.

**smartctl** optionen device

smartctl steuert die SMART-Funktionen der Festplatte. Damit können Sie mögliche Probleme der Festplatte frühzeitig erkennen.

- ▶ `-a` bzw. `--all`  
liefert alle verfügbaren SMART-Informationen.
- ▶ `-d ata/scsi/3ware,n`  
gibt den Festplattentyp an. Die Option ist nur erforderlich, wenn smartctl den Typ nicht selbst erkennen kann. Das gilt insbesondere für SATA-Festplatten, bei denen smartctl oft glaubt, es würde sich um SCSI-Festplatten handeln. Die Einstellung `3ware,n` ermöglicht die Nutzung der SMART-Funktionen für Festplatten, die an einen 3ware-RAID-Controller angeschlossen sind (zu den Details siehe man smartctl).
- ▶ `-H` bzw. `--health`  
gibt an, ob die Festplatte momentan in Ordnung ist und voraussichtlich die nächsten 24 Stunden noch funktionieren wird.
- ▶ `-i` bzw. `--info`  
liefert Basisinformationen zur Festplatte.
- ▶ `-l error/selftest` bzw. `--log=error/selftest`  
liefert Informationen zu den letzten fünf Fehlern bzw. zu den Ergebnissen der letzten 21 Selbsttests.
- ▶ `-s on/off` bzw. `--smart=on/off`  
aktiviert bzw. deaktiviert die SMART-Funktionen.
- ▶ `-t short/long` bzw. `--test=short/long`  
führt einen kurzen bzw. langen Selbsttest durch. Das Testergebnis sehen Sie sich später mit `-l selftest` an.

## Beispiel

`smartctl -H` bzw. `smartctl --health` gibt an, ob die Festplatte momentan in Ordnung ist. Sollte `smartctl` hier nicht *PASSED* als Ergebnis liefern, sollten Sie *sofort* damit beginnen, ein komplettes Backup durchzuführen!

```
root# smartctl -H /dev/sda
...
SMART overall-health self-assessment test result: PASSED
```

Für die Festplatte `/dev/sda` wurden bisher drei Selbsttests ausgeführt: einer unmittelbar nach dem Einbau der Platte (nach 40 Betriebsstunden), die anderen beiden nach ca. 2600 Stunden. Es sind keine Probleme aufgetreten.

```
root# smartctl -t short /dev/sda
root# smartctl -t long /dev/sda
...
root# smartctl -l selftest /dev/sda
```

Num	Test_Description	Status	Remaining	LifeTime	LBA
# 1	Extended offline	Completed without error	00%	2592	-
# 2	Short offline	Completed without error	00%	2591	-
# 3	Short offline	Completed without error	00%	40	-

```
smbclient [optionen]
smbclient verzeichnis [password] [optionen]
```

`smbclient` ermöglicht einen interaktiven Zugriff auf Windows-Netzwerkverzeichnisse. In der ersten Syntaxvariante ermitteln Sie mit dem Programm verfügbare Windows-Ressourcen. In der zweiten Variante geben Sie das gewünschte Netzwerkverzeichnis an (z.B. `//mars/data`, wenn Sie das Verzeichnis `data` auf dem Rechner `mars` nutzen möchten). Nach dem Verbindungsaufbau gelangen Sie in eine Shell. Anschließend können Sie wie beim Kommando `ftp` Verzeichnisse mit `ls` ansehen, mit `cd` wechseln, mit `get` Dateien auf den lokalen Rechner übertragen (*download*) und mit `put` Dateien auf dem externen Rechner speichern (*upload*). Einen Überblick über die wichtigsten Kommandos bekommen Sie mit `help`.

### ► `-L rechnername`

liefert die Liste der Netzwerkressourcen auf dem angegebenen Windows-Rechner bzw. Samba-Server.

### ► `-N`

verzichtet auf die automatische Passwortabfrage. Die Option ist nur zweckmäßig, wenn Sie wissen, dass für den Zugriff auf eine bestimmte Netzwerkressource kein Passwort erforderlich ist.



► `-U benutzername`

gibt den Benutzernamen an (nur notwendig, wenn dieser vom aktuellen Login-Namen abweicht).

► `-W arbeitsgruppenname`

gibt den Namen der Windows-Workgroup an (nur notwendig, wenn diese nicht in `/etc/samba/smb.conf` enthalten ist).

## Beispiel

Zuerst stellt `smbclient` eine Verbindung zur NAS-Festplatte her, anschließend liefert `ls` die Liste der dort gespeicherten Verzeichnisse und Dateien:

```
user$ smbclient -U name -W wgroupname '\\mynas\myshare'
Password: xxxxxx
Domain=[wgroupname] OS=[Unix] Server=[Samba 3.5.2]
smb: > ls
      data                D           0   Wed Apr  5 18:17:11 2014
      file.xy             AR          226  Sat Dec 14 00:00:00 2009
      ...
```

**smbpasswd** [optionen] [name]

`smbpasswd` verändert das Samba-Passwort des aktiven bzw. des angegebenen Benutzers.

► `-a`

erzeugt einen neuen Samba-Account. Es muss einen gleichnamigen Linux-Account geben.

► `-n`

entfernt das Passwort für den angegebenen Account. Die Netzwerkverzeichnisse sind nun ohne Passwort erreichbar, falls der Samba-Server entsprechend konfiguriert ist (`null passwords = yes`).

► `-x`

löscht den angegebenen Samba-Account.

**smbtree** [optionen]

`smbtree` liefert eine baumförmige Aufstellung aller im Netzwerk erreichbaren Windows-Netzwerkverzeichnisse bzw. Samba-Objekte.

► -b

durchsucht das Netzwerk mit *Broadcast Requests*, anstatt sich auf die Ergebnisse des lokalen Master-Browsers zu verlassen.

► -dn

gibt den Debug-Level zwischen 0 (keine Debugging-Informationen) und 10 (maximales Logging) an. Zur Fehlersuche führen Sie `smbtree` mit der Option `-d10` aus. Sie erhalten dann alle möglichen Debugging-Ausgaben.

Mitunter scheitert die Verbindung zu Windows-Netzwerkverzeichnissen daran, dass es im Netzwerk keinen Computer bzw. kein Gerät gibt, das als *Master-Browser* agiert. Abhilfe kann dann entweder ein *echter* Windows-Rechner schaffen oder natürlich ein Linux-Rechner, auf dem Samba läuft.

► -D

liefert nur eine Liste der erreichbaren Domänen bzw. Arbeitsgruppen.

► -N

verzichtet auf die Passwortabfrage. Das ist nur sinnvoll, wenn es Netzwerkressourcen gibt, die ohne Passwort erreichbar sind.

► -S

liefert nur eine Liste der erreichbaren Server.

► -U bzw. `--user=name[%password]`

gibt den Namen (und optional das Passwort) an.

### Beispiel

Das folgende Kommando liefert eine Liste aller Windows- bzw. Samba-Netzwerkressourcen, auf die der aktuelle Benutzer zugreifen kann:

```
user$ smbtree
Enter user's password:
WORKGROUP
    \\WD-NAS
        \\WD-NAS\Public
        \\WD-NAS\Download
        \\WD-NAS\multimedia
        \\WD-NAS\archive
    ...
    My Book World Edition Network Storage
        Public Share
        Download Share
```

```
snapper [optionen] kommando
```

SUSE-Distributionen verwenden standardmäßig ein `btrfs`-Dateisystem für die Systempartition. YaST erstellt automatisch vor und nach jedem administrativen Vorgang einen Snapshot. Diese Snapshots können wahlweise mit dem YaST-Modul `SNAPPER` oder mit dem hier kurz vorgestellten Kommando `snapper` verwaltet werden. Dem auszuführenden `snapper`-Kommando können Sie eine oder mehrere Optionen voranstellen:

- ▶ `--iso`  
formatiert Zeiten im ISO-Format.
- ▶ `--utc`  
zeigt UTC-Zeiten an (Universal Time Coordinated).

Die Aktionen, die eigentlich ausgeführt werden sollen, werden durch Subkommandos angegeben:

- ▶ `cleanup number|timeline|empty-pre-post`  
löscht alte Snapshots, wobei als Kriterium wahlweise eine Maximalanzahl von Snapshots, deren Alter oder deren Inhalt verwendet wird (siehe auch die Konfigurationsdatei `/etc/snapper/configs/root`).
- ▶ `create`  
erzeugt einen neuen Snapshot.
- ▶ `delete n bzw. n1-n2`  
löscht die angegebenen Snapshots. Die zu verwendenden Nummern gehen aus `snapper list` hervor.
- ▶ `list`  
listet alle Snapshots auf.
- ▶ `rollback [n]`  
macht den angegebenen oder den aktuellsten verfügbaren Snapshot zum aktiven Snapshot (genau genommen: zum `btrfs`-Default-Subvolume). Ein anschließendes `reboot`-Kommando versetzt das System in den Zustand, den es zum Zeitpunkt der Snapshot-Erstellung innehatte.

- ▶ `undochange n1..n2 [dateien]`

macht die zwischen dem Snapshot *n1* und dem Snapshot *n2* durchgeführten Änderungen rückgängig – wahlweise für alle betroffenen Dateien oder nur für die als Parameter angegebenen Dateien.

### **sort** [optionen] *datei*

`sort` sortiert die angegebene Datei und gibt das Ergebnis auf dem Bildschirm aus. Die Sortierordnung ist durch die Spracheinstellungen und speziell durch die Umgebungsvariable `LC_COLLATE` definiert.

- ▶ `-c`

überprüft, ob die Datei sortiert ist oder nicht.

- ▶ `-f`

behandelt Klein- und Großbuchstaben als gleichwertig.

- ▶ `-k n1,n2`

berücksichtigt nur die Zeichen zwischen der *n1*-ten und der *n2*-ten Spalte. Die Spaltennummerierung beginnt mit 1. Spalten werden normalerweise durch Leer- oder Tabulatorzeichen voneinander getrennt, siehe aber `-tz`. Wenn *n2* nicht angegeben wird, dann werden alle Zeichen ab *n1* bis zum Zeilenende berücksichtigt.

- ▶ `-m`

fügt zwei oder mehrere vorsortierte Dateien zu einer großen sortierten Datei zusammen: Das geht schneller, als die Dateien vorher zusammenzufügen und erst dann zu sortieren.

- ▶ `-o ergebnisdatei`

schreibt das Ergebnis in die angegebene Datei. Die Datei darf mit der zu sortierenden Datei übereinstimmen.

- ▶ `-r`

sortiert in umgekehrter Reihenfolge.

- ▶ `-tz`

gibt das Trennzeichen zwischen zwei Spalten an. Die Standardeinstellung ist *white space*, also eine beliebige Mischung aus Leer- und Tabulatorzeichen.

## Beispiel

Das folgende Kommando sortiert das von `ls` übergebene Inhaltsverzeichnis nach Anwender- und Gruppennamen. Dateien mit übereinstimmenden Anwender- und Gruppennamen werden ihrer Größe nach sortiert.

```
user$ ls -l | sort -k 3
```

### source datei

`source` führt in einem `bash`-Script die angegebene Datei so aus, als befänden sich die darin enthaltenen Kommandos an der Stelle des `source`-Kommandos. Nach der Ausführung der Datei wird das laufende Shell-Programm in der nächsten Zeile fortgesetzt. Zur Ausführung der angegebenen Datei wird keine neue Shell gestartet. Alle Variablen inklusive der Parameterliste gelten daher auch für die angegebene Datei. Wenn in dieser Datei `exit` ausgeführt wird, kommt es *nicht* zu einem Rücksprung in das Programm mit dem `source`-Kommando, sondern zu einem sofortigen Ende der Programmausführung.

Zu `source` existiert die Kurzform `._datei`. Dabei steht `_` für ein Leerzeichen.

### speaker-test [optionen]

Das Kommando `speaker-test` aus dem Paket `alsa-utils` hilft dabei, Problemen des Audio-Systems auf die Spur zu kommen. Dazu generiert es Testtöne, die über das Audio-System ausgegeben werden.

- ▶ `-c n`  
verwendet  $n$  Audio-Kanäle.
- ▶ `-f n`  
spielt einen Sinus-Ton in der angegebenen Frequenz in Hertz. Gleichzeitig muss die Option `-t sine` verwendet werden.
- ▶ `-s n`  
testet nur den Audio-Kanal mit der Nummer  $n$ .
- ▶ `-t pink|sine|wav`  
gibt rosa Rauschen, eine Sinuswelle oder ein Sprechmuster aus einer WAV-Datei aus.

► `-w datei`

spielt die angegebene WAV-Datei ab. Gleichzeitig muss die Option `-t wav` verwendet werden.

**Beispiel**

Das folgende Kommando gibt über den linken und den rechten Audio-Kanal die Wörter *front left* bzw. *front right* aus. Das Sprechmuster wird aus einer WAV-Datei geladen. Die Ausgabe endet erst mit `[Strg]+[C]` wieder.

```
user$ speaker-test -t wav -c2
```

**split** [optionen] datei [zieldatei]

`split` zerlegt die angegebene Datei in mehrere Einzeldateien. Die Ausgangsdatei wird dabei (je nach Optionen) alle  $n$  Byte, kByte oder Zeilen getrennt. Wenn keine Zieldatei angegeben wird, liefert das Kommando die Dateien `xaa`, `xab` etc. Wenn eine Zieldatei angegeben wird, dann wird für die Ergebnisdateien dieser Dateiname zusammen mit der Zeichenkombination `aa`, `ab` etc. verwendet.

► `-n` bzw. `-l n` bzw. `--lines=n`

zerlegt die Ausgangsdatei in Einzeldateien mit je  $n$  Zeilen.

► `-b n` bzw. `--bytes=n`

trennt die Ausgangsdatei alle  $n$  Bytes. Die Größenangabe kann auch in kByte, MByte, GByte bzw. in Vielfachen von 1000 Byte angegeben werden: `kB` (1000 Byte), `K` (1024 Byte), `MB` ( $10^6$  Byte), `M` ( $1024^2$  Byte), `GB` ( $10^9$  Byte oder  $G$  ( $1024^3$  Byte).

► `-C n` bzw. `--line-bytes=n`

wie `-b`, allerdings werden die Dateien an Zeilengrenzen zerlegt und sind daher zumeist einige Byte kleiner als  $n$ .

**Beispiel**

Im folgenden Beispiel zerlegt `split` die Archivdatei `backup.tgz` in Einzeldateien zu je 4 GByte und benennt sie mit `dvd.aa`, `dvd.ab` etc. Diese Dateien könnten anschließend auf DVDs gebrannt werden.

```
user$ split -C 4G backup.tgz dvd.
```

`cat` setzt die Einzeldateien wieder zu einer Gesamtdatei zusammen:

```
user$ cat dvd.* > total.tar
```

```
sqlite3 [optionen] datenbankdatei
sqlite3 [optionen] datenbankdatei 'sqlkommando1;sql2;...'
```

sqlite3 wird normalerweise als Shell zur Bearbeitung von SQLite-Datenbanken genutzt. Nach dem Start können Sie interaktiv SQL-Kommandos ausführen, wobei alle Kommandos mit einem Semikolon abgeschlossen werden müssen. Neben den üblichen SQL-Kommandos kennt sqlite3 einige spezifische Kommandos, die mit einem Punkt beginnen, z. B. `.help` zur Anzeige eines Hilfetexts oder `.tables` zur Auflistung aller Tabellen in der Datenbank. `.quit` oder `[Strg]+[D]` beenden sqlite3.

In der zweiten Syntaxvariante führt sqlite3 die angegebenen SQL-Kommandos direkt aus.

### Beispiel

Das folgende Kommando ermittelt, welche Musikalben in einer ownCloud-Datenbank gespeichert sind:

```
user$ sqlite owncloud.db
sqlite> select * from oc_media_albums;
1|Nightclub|1|
2|The Monroe Doctrine|2|
3|The Best Of Herbie Mann|3|
...
sqlite> .quit
```

```
ss [optionen]
```

`ss` (*socket statistics*) analysiert die Netzwerkaktivität des lokalen Rechners. Wenn das Kommando ohne Optionen ausgeführt wird, liefert es eine Liste aller *non-listening* TCP-Sockets mit einer aktiven Verbindung. `ss` bietet eine ähnliche Funktionalität wie `netstat`, gilt aber als das modernere Kommando und liefert in manchen Fällen detailliertere Informationen.

- ▶ `-4` bzw. `-6`  
berücksichtigt nur IPv4- bzw. nur IPv6-Verbindungen.
- ▶ `-a`  
zeigt alle Sockets an, sowohl *listening* als auch *non-listening*.
- ▶ `-e`  
liefert noch mehr Details (*extended*).

- ▶ `-n`  
zeigt numerische Portnummern an, nicht die Namen der Dienste.
- ▶ `-p`  
gibt an, welche Prozesse welche Sockets nutzen.
- ▶ `-r`  
ersetzt Portnummern durch die Namen der Dienste.
- ▶ `-s`  
zeigt nur zusammenfassende Informationen an (*summary*).
- ▶ `-t` bzw. `-u`  
berücksichtigt nur TCP- bzw. nur UDP-Verbindungen.

### Beispiel

Das folgende Kommando liefert eine Liste aller aktiven TCP-Sockets für IPv6-Verbindungen:

```
user$ ss -6 -t -a
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
LISTEN	0	100	:::pop3	:::*
LISTEN	0	100	:::imap2	:::*
LISTEN	0	128	:::http	:::*
LISTEN	0	100	:::smtp	:::*
LISTEN	0	128	:::https	:::*
LISTEN	0	100	:::imaps	:::*
LISTEN	0	100	:::pop3s	:::*
TIME-WAIT	0	0	::ffff:5.9.22.28:http	::ffff:188.192.181.195:50045
TIME-WAIT	0	0	::ffff:5.9.22.28:http	::ffff:188.192.181.195:50047
ESTAB	0	46200	::ffff:5.9.22.28:http	::ffff:46.10.92.30:22893
FIN-WAIT-1	0	1	::ffff:5.9.22.28:https	::ffff:203.134.55.243:51106
FIN-WAIT-2	0	0	::ffff:5.9.22.28:http	::ffff:46.10.92.30:24262
...				

```
ssh [optionen] rechnername
ssh [optionen] rechnername kommando
```

In der ersten Syntaxvariante öffnet `ssh` eine Shell auf einem anderen Rechner und ermöglicht es so, auf diesem Rechner interaktiv zu arbeiten. Die zweite Syntaxvariante führt ein einzelnes Kommando auf dem zweiten Rechner aus. `ssh` funktioniert nur, wenn auf dem zweiten Rechner ein SSH-Server läuft.



► `-l loginname`

verwendet auf dem zweiten Rechner den angegebenen Login-Namen statt des aktuellen Benutzernamens. Der gewünschte Login-Name kann auch in der Form `ssh loginname@rechnername` angegeben werden.

► `-L localport:host:hostport`

bildet einen Tunnel zwischen dem lokalen Rechner (Port `localport`) und dem zweiten Rechner (Port `hostport`).

► `-v`

liefert umfassende Debugging-Informationen. Wenn Sie Login-Probleme haben, hilft `ssh -v` oft dabei, die Ursache herauszufinden. Die Option kann bis zu dreimal angegeben werden (also `-v -v -v`), `ssh` liefert dann noch mehr Debugging-Informationen.

► `-X`

ermöglicht die Ausführung von X-Programmen.

## Beispiele

Das folgende Kommando zeigt, wie Sie sich auf einem externen Rechner einloggen, wobei der Benutzername unverändert bleibt. Anschließend können Sie auf dem externen Rechner (z. B. Ihrem Root-Server) Kommandos ausführen.

```
kofler@uranus$ ssh kofler.info
kofler@kofler.info's password: *****
```

Wenn Sie mit `ssh` zum ersten Mal eine Verbindung zu einem anderen Rechner herstellen, erscheint oft eine Warnung nach dem folgenden Muster:

```
The authenticity of host 'kofler.info (1.2.3.4)' can't be established.
RSA1 key fingerprint is 1e:0e:15:ad:6f:64:88:60:ec:21:f1:4b:b7:68:f4:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'kofler.info,1.2.3.4' to the list of known hosts.
```

Das folgende Kommando zeigt, wie ich den gesamten `/var/www`-Verzeichnisbaum meines Webservers `kofler.info` in das lokale Verzeichnis `~/bak` kopiere. Dabei wird `ssh` dazu genutzt, ein einzelnes Kommando (`tar`) auszuführen. Das setzt voraus, dass `kofler@kofler.info` alle Dateien aus `/var/www` lesen kann. Grundsätzlich wäre es auch möglich, das `tar`-Kommando mit `ssh -l root` mit root-Rechten auszuführen. Allerdings sind die meisten SSH-Server aus Sicherheitsgründen so konfiguriert, dass ein direkter root-Login unmöglich ist.

Die Ergebnisse von tar werden via SSH abhörsicher auf den lokalen Rechner übertragen und vom zweiten tar-Kommando weiterverarbeitet.

```
kofler@uranus$ ssh kofler.info tar -cf - /var/www | tar -xC ~/bak/ -f -
kofler@kofler.info's password: ******
```

**ssh-copy-id** [optionen] user@host

ssh-copy-id fügt den öffentlichen Schlüssel des lokalen Rechners in die Datei `.ssh/authorized_keys` des Host-Rechners ein. Wenn es mehrere Schlüssel gibt, entscheidet das Kommando `ssh-add -L`, welcher Schlüssel übertragen wird.

► `-i [schlüsseldatei]`

überträgt die angegebene Schlüsseldatei bzw. `.ssh/id_rsa.pub`.


Sollte das Script `ssh-copy-id` nicht zur Verfügung stehen oder die Schlüsselübertragung an den Sicherheitseinstellungen des externen SSH-Servers scheitern, muss die öffentliche Schlüsseldatei (in der Regel `.ssh/id_rsa.pub`) manuell zum Server übertragen und dort der Datei `.ssh/authorized_keys` hinzugefügt werden.

Wenn auf dem Host-System Fedora, RHEL oder ein RHEL-Derivat läuft (CentOS etc.), kann der Einsatz von `ssh-copy-id` dazu führen, dass das Verzeichnis `.ssh` und die Datei `authorized_keys` ohne die SELinux-Kontextinformationen erzeugt wird. Abhilfe schafft eines der beiden folgenden Kommandos auf dem Hostrechner:

```
root# /sbin/restorecon -r /root/.ssh
root# /sbin/restorecon -r /home/username/.ssh
```

**ssh-keygen** [optionen]

ssh-keygen erzeugt, verwaltet und konvertiert kryptografische Schlüssel, die zur Authentifizierung mit SSH verwendet werden können. Wenn `ssh-keygen` ohne weitere Optionen aufgerufen wird, erzeugt es ein RSA-Schlüsselpaar mit einer Schlüssellänge von 2048 Bits. Der private und der öffentliche Teil des Schlüssels wird in den Dateien `.ssh/id_rsa` und `.ssh/id_rsa.pub` gespeichert.

Der Zugriff auf den Schlüssel wird standardmäßig durch ein Passwort abgesichert; wenn Sie statt der Passworтеingabe einfach  drücken, entfällt dieses Sicherheitsmerkmal. Wenn ein privater, nicht durch ein Passwort abgesicherter Schlüssel einem Dritten in die Hände fällt, kann dieser Dritte sich ohne Weiteres auf allen Rechnern anmelden, auf denen Sie den öffentlichen Teil des Schlüssels installiert haben!

Durch zusätzliche Optionen können Sie beim Erzeugen neuer Schlüssel unterschiedliche Verschlüsselungsverfahren, Schlüssellängen und Speicherorte angeben, vorhandene Schlüssel konvertieren etc.

### **start/stop** dienst

**start** startet einen durch Upstart verwalteten Hintergrundprozess, **stop** beendet den Prozess. *name* muss mit dem Namen einer Upstart-Konfigurationsdatei übereinstimmen. Universeller als **start/stop** ist service dienst kommando.

### **stat** [optionen] dateien

**stat** liefert ausführliche Informationen über die angegebenen Dateien, unter anderem die Zugriffsrechte in der **ls**-Schreibweise und als Oktalzahl, die Anzahl der genutzten Blöcke, den Zeitpunkt der letzten Änderung und des letzten Lesezugriffs etc.

► **-c formatstring**

formatiert die Ausgabe entsprechend der Formatzeichenkette.

► **-f**

liefert Informationen über das Dateisystem, in dem sich die Datei befindet (anstelle von Informationen über die Datei). In der Zeichenkette können unter anderem die folgenden Codes verwendet werden:

- %a: Zugriffsrechte in Oktalschreibweise
- %A: Zugriffsrechte in **ls**-Schreibweise
- %C: SELinux-Kontext
- %F: Dateityp (Datei, Verzeichnis, Block-Device etc.)
- %g: GID-Nummer der Gruppe, der die Datei zugeordnet ist
- %G: Name der Gruppe, der die Datei zugeordnet ist
- %h: Anzahl der Hard-Links
- %n: Dateiname
- %s: Dateigröße in Bytes
- %u: UID-Nummer des Dateibesitzers
- %U: Name des Dateibesitzers

## Beispiel

Ohne Optionen liefert `stat` eine übersichtliche Zusammenstellung diverser Metadaten einer Datei. Mit `-c` kann das Ergebnis auf Einzelinformationen reduziert werden, im zweiten Beispiel auf die oktalen Zugriffsrechte und den Dateinamen.

```
user$ stat druck.pdf
Datei: druck.pdf
Größe: 390758      Blöcke: 768      EA Block: 4096   reguläre Datei
Gerät: fd01h/64769d  Inode: 397971   Verknüpfungen: 1
Zugriff: (0664/-rw-rw-r--) Uid: ( 1000/  kofler)  Gid: ( 1000/  kofler)
Kontext: unconfined_u:object_r:user_home_t:s0
Zugriff   : 2013-09-24 11:35:17.316000000 +0200
Modifiziert: 2013-08-06 17:53:16.083000000 +0200
Geändert  : 2013-08-06 17:53:16.083000000 +0200
Geburt    : -
user$ stat -c "%a %n" druck.pdf
664 druck.pdf
```

**strace** [optionen] kommando

`strace` führt das angegebene Kommando oder Programm aus und zeigt alle Systemfunktionen an, die das Kommando aufruft. `strace` ist damit ein wichtiges Werkzeug zur Fehlersuche in kompilierten Programmen. Durch diverse Optionen kann gesteuert werden, welche (Zusatz-)Informationen das Programm ausgeben soll.

**strings** [optionen] datei

Das Kommando extrahiert Zeichenketten aus einer Binärdatei und zeigt diese an. Das ist beispielsweise dann praktisch, wenn Sie einen Text (z. B. eine Fehlermeldung oder einen Dateinamen) in einer Programmdatei suchen.

## Beispiele

Das folgende Kommando ermittelt alle Zeichenketten, die sich in der Binärdatei `/bin/ls` befinden und das Wort `error` enthalten:

```
root# strings /bin/ls | grep error
error
error initializing month strings
write error
```

Im folgenden Beispiel wollte ich ergründen, ob bzw. auf welche Tastaturkonfigurationsdatei(en) das Systemd-Programm `systemd-vconsole-setup` unter openSUSE Leap zugreift:

```
user$ strings /usr/lib/systemd/systemd-vconsole-setup | grep keyboard
/etc/sysconfig/keyboard
Failed to read /etc/sysconfig/keyboard: %s
```

```
su [optionen] [user]
```

su (*substitute user*) ohne Optionen wechselt mit Passwordeingabe in den root-Modus. Damit ist bis zum nächsten exit-Kommando root der aktive User.

Optional kann bei su statt des Standard-Users root auch ein anderer User angegeben werden. Wenn su von root ausgeführt wird, muss beim User-Wechsel nicht einmal ein Passwort angegeben werden.

► -c '*kommando*'

führt nur das angegebene Kommando mit root-Rechten aus.

► -l bzw. --login

beim User-Wechsel wird die neue Shell als Login-Shell gestartet. Das heißt, es werden alle Login-Dateien mit eingelesen. Das ist erforderlich, damit Umgebungsvariablen wie PATH korrekt konfiguriert werden.

```
sudo [optionen] [var1=wert1 var2=wert2] kommando
```

sudo führt ein Kommando aus, als würde es von einem anderen Benutzer (normalerweise root) ausgeführt. Damit können dank sudo gewöhnliche Benutzer administrative Aufgaben übernehmen bzw. systemkritische Kommandos ausführen, ohne dazu das root-Passwort zu kennen.

Bevor sudo die Ausführung eines Programms erlaubt, muss dieses Recht für einen bestimmten Benutzer und für ein bestimmtes Programm in der Datei /etc/sudoers angegeben werden.

► -b

startet das angegebene Kommando im Hintergrund.

► -H

trägt das Home-Verzeichnis des Benutzers, für dessen Account das Kommando ausgeführt wird, in die \$HOME-Umgebungsvariable ein (also üblicherweise /root).

► -K

löscht das gespeicherte Passwort. Beim nächsten Aufruf von sudo muss neuerlich das Passwort angegeben werden.

► `-s`

startet die in `$SHELL` angegebene Shell. Das ermöglicht es, mehrere Kommandos in einer Shell auszuführen, ohne jedes Mal `sudo` voranzustellen. Die Shell kann mit `exit` oder `[Strg]+[D]` verlassen werden.

► `-u benutzer`

startet das Kommando für den angegebenen Benutzer (statt für `root`).

Im `sudo`-Kommando definierte Variablen werden als Umgebungsvariablen weitergegeben. Daher ist `sudo var1=x var2=y cmd` eine Kurzschreibweise für die folgenden Kommandos:

```
user$ sudo -s
root# export var1=x
root# export var2=y
root# cmd
root# exit
```

### Beispiel

Im folgenden Beispiel wird `sudo` dazu verwendet, das Kommando `apt-get install` mit `root`-Rechten auszuführen:

```
user$ sudo apt-get install gimp
[sudo] password for user: *****
...
```

### svn kommando

Das Kommando `svn` aus dem Paket `subversion` steuert das gleichnamige Programm zur Versionsverwaltung. Subversion wird häufig eingesetzt, wenn mehrere Personen zugleich ein Programm entwickeln. Subversion verwaltet ein zentrales Codearchiv (das Subversion-Repository auf dem Subversion-Server) und ermöglicht es, jede Änderung am Code zurückzuverfolgen und bei Bedarf rückgängig zu machen.

► `add [-N] datei/verzeichnis`

fügt die Datei der Subversion-Kontrolle hinzu. Die Datei wird mit dem nächsten SVN-Commit in das Subversion-Repository übertragen. Bei Verzeichnissen bewirkt `-N`, dass der Inhalt des Verzeichnisses *nicht* berücksichtigt wird.

► `checkout/co [--username name] url`

überträgt erstmalig alle Dateien vom SVN-Server in das aktuelle Verzeichnis auf dem lokalen Rechner. Dabei muss in der Regel ein Passwort angegeben werden (je

nach Konfiguration des SVN-Servers). `svn` speichert einen Hash des Passworts in `.subversion/auth/*`, sodass die Passwortangabe in Zukunft nicht wiederholt werden muss.

- ▶ `commit/ci -m 'kommentar'`

lädt die zuletzt durchgeführten Änderungen in das Subversion-Repository hoch. Vor jedem Commit sollte `svn update` ausgeführt werden. Wenn Sie vor dem Commit überprüfen möchten, welche Änderungen gespeichert werden, führen Sie vorher `svn status` aus.

- ▶ `copy oldurl newurl -m 'kommentar'`

erzeugt eine Kopie des SVN-Verzeichnisses, wobei die Kopie unter der Kontrolle von Subversion bleibt. Kopien werden häufig zur Verwaltung von Zweigen (*branches*) verwendet. Es ist üblich, das Unterverzeichnis *branches* zur Speicherung der Zweige zu verwenden.

- ▶ `diff -r n1:n2 [datei]`

zeigt an, wie sich die angegebene Datei von Revision *n1* bis zur Revision *n2* verändert hat. Wenn keine Datei angegeben wird, ermittelt `svn diff` die Auflistung *aller* Änderungen.

- ▶ `diff -r url1[@n1] url2[@n2]`

ermittelt die Unterschiede zwischen zwei SVN-Zweigen. Wenn keine Revisionsnummern angegeben werden, berücksichtigt `svn` jeweils die aktuellste Version jeden Zweigs.

- ▶ `help [kommando]`

liefert eine Liste aller SVN-Kommandos bzw. einen detaillierten Hilfetext zum angegebenen Kommando.

- ▶ `list/ls url`

liefert eine Liste aller unter SVN-Kontrolle befindlichen Dateien auf dem SVN-Server. Das Kommando ist vor allem dann zweckmäßig, wenn Sie die Dateien nicht herunterladen möchten.

- ▶ `log [datei]`

listet alle SVN-Commits auf (die letzte Änderung zuerst). Wenn eine Datei angegeben wird, liefert `svn log` nur solche Commits, durch die diese Datei verändert wurde.

- ▶ `move/mv/ren datei1 datei2`

gibt einer unter SVN-Kontrolle stehenden Datei einen neuen Namen.

- ▶ `revert datei`  
widerruft alle Änderungen, die seit dem letzten SVN-Update an der lokalen Datei durchgeführt wurden.
- ▶ `rm datei`  
entfernt die Datei aus Subversion. Die Datei wird mit dem nächsten SVN-Commit auf dem Subversion-Server gelöscht.
- ▶ `status/st`  
fasst die Änderungen zusammen, die seit dem letzten SVN-Update an den lokalen Dateien vorgenommen wurden.
- ▶ `switch url`  
aktiviert den Zweig (*branch*), den Sie durch seine URL angeben.
- ▶ `update/up`  
aktualisiert die lokalen Dateien mit der neuesten Version vom Subversion-Server.
- ▶ `update -r n datei`  
ersetzt die angegebene Datei durch eine ältere Version (Revision *n*).

### Beispiel

Die folgenden Zeilen zeigen, wie Sie einem Projekt eine neue PHP-Datei hinzufügen:

```
user$ svn update
user$ svn add modules/html.php
user$ svn commit -m 'neues HTML-Modul'
```

```
swapon device
swapoff device
```

`swapon` aktiviert das angegebene Device (zumeist eine Festplattenpartition) bzw. die stattdessen angegebene Datei als Swap-Bereich. Die Partition bzw. Datei muss vorher mit `mkswap` als Swap-Bereich formatiert werden. `swapon` wird beim Hochfahren von Linux für alle in `/etc/fstab` aufgezählten Swap-Bereiche automatisch ausgeführt. Ein Beispiel zu `swapon` finden Sie bei der Beschreibung von [mkswap](#).

`swapoff` deaktiviert die angegebene Swap-Datei oder -Festplattenpartition.



**sync**

Führt alle gepufferten Schreiboperationen auf den Festplatten aus. Wenn aus irgendeinem Grund ein geordnetes Beenden von Linux nicht möglich ist – d. h., wenn die Kommandos shutdown, reboot und halt nicht ausführbar sind und der Rechner nicht auf `[Alt]+[Strg]+[Entf]` reagiert –, dann sollte `sync` unmittelbar vor dem Ausschalten ausgeführt werden. Das ist aber nur eine Notlösung!

**sysctl** optionen

`sysctl` liest den Zustand von Kernelparametern bzw. verändert ihren Wert im laufenden Betrieb.

- ▶ `-a`  
liefert eine Liste aller verfügbaren Kernelparameter inklusive der aktuellen Einstellung.
- ▶ `-n parameter`  
liefert den aktuellen Wert des angegebenen Kernelparameters.
- ▶ `-p [dateiname]`  
liest `/etc/sysctl.conf` bzw. die angegebene Datei und verändert die dort angegebenen Kernelparameter entsprechend.
- ▶ `-w parameter=wert`  
verändert den angegebenen Parameter.

**Beispiel**

Die beiden folgenden Kommandos aktivieren zuerst die Forwarding-Funktionen des Kernels und dann das Masquerading für die Netzwerkschnittstelle `eth0`:

```
root# sysctl -w net.ipv4.ip_forward=1
root# iptables -A POSTROUTING -t nat -o eth0 -j MASQUERADE
```

**systemctl** kommando

`systemctl` dient zur Administration des Init-Prozess bei Distributionen, die Systemd als Init-System verwenden. Das ist bei nahezu allen aktuellen Distributionen der Fall. Sie können mit `systemctl` eine Liste aller durch Systemd verwalteten Prozesse ermitteln, einzelne Hintergrunddienste beenden oder neu starten etc.

systemctl unterscheidet unter anderem zwischen gewöhnlichen Diensten (services), Sockets, Devices, Mount-Diensten und Zielen (targets, vergleichbar mit Init-V-Runleveln).

- ▶ `enable/disable name`  
aktiviert den angegebenen Dienst dauerhaft bzw. deaktiviert ihn wieder. Bei der Ausführung des Kommandos werden die entsprechenden Links eingerichtet bzw. wieder entfernt. Bei aktuellen systemd-Versionen bewirkt die zusätzliche Option `--now`, dass der betreffende Dienst auch gleich gestartet bzw. gestoppt wird.
- ▶ `get-default`  
ermittelt das Default-Target.
- ▶ `halt/poweroff/reboot`  
fährt das System herunter bzw. startet es neu.
- ▶ `is-active name`  
testet, ob der angegebene Dienst aktiv ist.
- ▶ `isolate name.target`  
aktiviert das angegebene Target sowie alle davon abhängigen Dienste. Nicht benötigte Dienste werden gestoppt. `isolate` hat damit eine ähnliche Funktion wie ein Runlevel-Wechsel bei einem Init-V-System.
- ▶ `list-timers`  
gibt einen Überblick über aktive Timer, also periodisch durch Systemd auszuführende Jobs. Bei jedem Job wird der Zeitpunkt der letzten und der geplanten nächsten Ausführung angegeben.
- ▶ `list-units`  
liefert eine Liste aller von Systemd verwalteten Dienste, Sockets, Targets etc. Mehrseitige Ausgaben werden durch `less` geleitet, sodass sie seitenweise gelesen werden können. Wenn Sie das nicht wünschen, geben Sie die Option `--no-pager` an. Das Ergebnis kann mit `--type=...` gefiltert werden. Zulässige Unit-Typen sind `service`, `socket`, `target`, `device`, `mount`, `automount` und `snapshot`.
- ▶ `reload/restart name`  
fordert den Prozess zum Neueinlesen der Konfiguration auf bzw. startet den Dienst gänzlich neu.
- ▶ `set-default name.target`  
legt das Default-Target für künftige Bootprozesse fest.

► `show name`

liefert detailliertere Daten als `status`. Das Ergebnis ist zeilenweise strukturiert und kann verhältnismäßig einfach durch ein Script verarbeitet werden.

► `start/stop name`

startet bzw. stoppt den angegebenen Init-Prozess bzw. Dienst.

► `status name`

liefert Statusinformationen zum angegebenen Dienst in einer gut lesbaren Form.

## Beispiele

Die folgenden Zeilen zeigen typische `systemctl`-Kommandos:

```
root# systemctl start ntpd.service (NTP-Dämon starten)
root# systemctl stop ntpd.service (NTP-Dämon stoppen)
root# systemctl restart ntpd.service (NTP-Dämon neu starten)
root# systemctl reload ntpd.service (Konfiguration des NTP-Dämons neu einlesen)
root# systemctl status ntpd.service (Status des NTP-Dämons ermitteln)
root# systemctl enable ntpd.service (NTP-Dämon in Zukunft automatisch starten)
ln -s '/lib/systemd/system/ntp.service'
    '/etc/systemd/system/multi-user.target.wants/ntp.service'
root# systemctl disable ntpd.service (NTP-Dämon nicht mehr automatisch starten)
rm '/etc/systemd/system/multi-user.target.wants/ntp.service'
root# systemctl isolate reboot.target (Neustart des Rechners)
```

## `tac` `datei`

`tac` gibt die angegebene Textdatei in umgekehrter Reihenfolge auf dem Bildschirm aus, d. h. die letzte Zeile zuerst. Der merkwürdige Kommandoname ergibt sich aus der Verdrehung der Buchstaben des `cat`-Kommandos, das Textdateien in richtiger Reihenfolge ausgibt.

## `tail` [`optionen`] `datei`

`tail` gibt die letzten zehn Zeilen einer Textdatei auf dem Bildschirm aus.

► `-n zeilen`

gibt die angegebene Anzahl von Zeilen aus.

► `-f`

liest die Datei regelmäßig aus und gibt alle neuen Zeilen aus. In dieser Form eignet sich `tail` vor allem zur Beobachtung von Protokolldateien.

## Beispiel

Das folgende Kommando zeigt die letzten zehn Zeilen von `messages` an. Wenn neue Zeilen hinzukommen, werden auch diese angezeigt, sodass nach kurzer Zeit der ganze Bildschirm genutzt wird – und nicht nur zehn Zeilen.

```
root# tail -f /var/log/messages
```

```
tar aktion [optionen] dateien  
tar aktion [optionen] verzeichnisse
```

`tar` vereint mehrere Dateien oder ganze Verzeichnisse in einem sogenannten Archiv bzw. extrahiert aus diesem Archiv wieder ihre Bestandteile. `tar` war ursprünglich als Tool zum Lesen und Schreiben von Daten auf einem Streamer konzipiert. `tar` greift deswegen standardmäßig auf den installierten Streamer (zumeist `/dev/tape` oder `/dev/rmt0`) zu. Wenn Sie ein Archiv in einer Datei anlegen möchten (beispielsweise, um diese Datei anschließend auf eine CD zu brennen), müssen Sie die Option `-f datei` angeben.

Da `tar` je nach Angabe der Optionen die zu archivierenden Dateien auch komprimiert, ist es in seiner Funktion mit dem unter Windows verbreiteten Programm WinZIP zu vergleichen. Die typische Kennung für Archivdateien ist `.tar`. Wenn die Archivdatei komprimiert ist, lauten die Kennungen zumeist `.tgz`, `.tar.gz`, `.tar.bz2` oder `.tar.xz`.

Die Steuerung von `tar` erfolgt in zwei Stufen: Zum einen muss eine Aktion angegeben werden, die `tar` ausführen soll, und zum anderen kann diese Aktion durch eine oder mehrere Optionen gesteuert werden. Auch wenn Aktionen und Optionen formal gleich aussehen, besteht ein wesentlicher Unterschied: Es muss genau eine Aktion vor allen anderen Optionen angegeben werden. Während in den nächsten Zeilen alle Aktionen kurz beschrieben sind, wurden hier nur die wichtigsten Optionen aufgezählt (siehe `man tar`).

Auf vielen Unix-Systemen erfolgt die Steuerung von `tar` zwar mit denselben Kommandos und Optionen, die Syntax ist aber anders: Es werden alle Kommandos und Optionen als ein Block ohne die üblichen Optionsstriche angegeben, etwa `tar cvf name.tar pfad`. GNU-`tar` versteht beide Syntaxvarianten, d. h., es ist Ihnen überlassen, ob Sie Optionsstriche verwenden oder nicht.

## Aktionen

- `-A` bzw. `--catenate` bzw. `--concatenate`

hängt an ein vorhandenes Archiv ein weiteres Archiv an. Diese Option ist nur für Streamer geeignet (nicht für Archivdateien).

- ▶ `-c` bzw. `--create`  
erzeugt ein neues Archiv, d. h., ein eventuell vorhandenes Archiv wird überschrieben.
- ▶ `-d` bzw. `--diff` bzw. `--compare`  
vergleicht die Dateien des Archivs mit den Dateien des aktuellen Verzeichnisses und stellt eventuelle Unterschiede fest.
- ▶ `--delete`  
löscht Dateien aus dem Archiv. Nur für Archivdateien (nicht für Streamer) geeignet.
- ▶ `-r` bzw. `--append`  
erweitert das Archiv um zusätzliche Dateien.
- ▶ `-t` bzw. `--list`  
zeigt das Inhaltsverzeichnis des Archivs an.
- ▶ `-u` bzw. `--update`  
erweitert das Archiv um neue oder geänderte Dateien. Die Option kann nicht für komprimierte Archive verwendet werden. Vorsicht: Das Archiv wird immer größer, weil bereits vorhandene Dateien nicht überschrieben werden! Die neuen Dateien werden einfach an das Ende des Archivs angehängt.
- ▶ `-x` bzw. `--extract`  
extrahiert die angegebenen Dateien aus dem Archiv und kopiert sie in das aktuelle Verzeichnis. Die Dateien werden dabei nicht aus dem Archiv gelöscht.

## Optionen

- ▶ `-C verzeichnis`  
extrahiert die Dateien in das angegebene (statt in das aktuelle) Verzeichnis.
- ▶ `-f datei`  
verwendet die angegebene Datei als Archiv (anstatt auf den Streamer zuzugreifen). Wenn statt eines Dateinamens ein einfacher Bindestrich angegeben wird (`-f -`), werden die Daten an den Standardausgabekanal weitergeleitet bzw. von der Standardeingabe gelesen. Das ist vor allem dann sinnvoll, wenn zwei `tar`-Kommandos durch `|` verknüpft werden sollen (typischerweise zum Transportieren ganzer Dateibäume).

- ▶ `-g timestamp-datei`  
speichert in der binären Timestamp-Datei Metadaten über die mit `tar` gesicherten Dateien und berücksichtigt bei weiteren Aufrufen von `tar` nur die Dateien, die sich seither verändert haben. Die Option bietet eine einfache Möglichkeit, inkrementelle Backups zu erzeugen (siehe auch das letzte der folgenden Beispiele).
- ▶ `-j` bzw. `--bzip2`  
komprimiert bzw. dekomprimiert das gesamte Archiv durch `bzip2` (siehe auch `-z!`).
- ▶ `-J` bzw. `--xz`  
komprimiert bzw. dekomprimiert das gesamte Archiv durch `xz`. Das setzt voraus, dass das Kommando `xz` zur Verfügung steht. Bei vielen Distributionen muss dazu das Paket `xz-utils` installiert werden.
- ▶ `-L n` bzw. `--tape-length n`  
gibt die Kapazität des Streamers in kByte an. Wenn die Kapazität durch die Größe des Archivs überschritten wird, fordert `tar` zu einem Wechsel des Magnetbands auf.
- ▶ `-N datum` bzw. `--after-date datum` bzw. `--newer datum`  
archiviert nur Dateien, die aktueller als das angegebene Datum sind. Damit lassen sich unkompliziert inkrementelle Backups erstellen.
- ▶ `-p` bzw. `--preserve-permissions`  
belässt beim Extrahieren der Dateien die Zugriffsrechte unverändert. Die Option gilt standardmäßig, wenn `tar` von `root` ausgeführt wird.
- ▶ `-T datei` bzw. `--files-from datei`  
archiviert bzw. extrahiert die in der Datei angegebenen Dateinamen.
- ▶ `-v` bzw. `--verbose`  
zeigt während der Arbeit alle Dateinamen auf dem Bildschirm an. Wenn `-v` in Kombination mit dem Kommando `t` verwendet wird, werden zusätzliche Informationen zu den Dateien angezeigt (Dateigröße etc.). Wenn die Option zweimal angegeben wird, werden die Informationen noch ausführlicher.
- ▶ `-W` bzw. `--verify`  
überprüft nach dem Schreiben die Korrektheit der gerade archivierten Dateien. Kann nicht für komprimierte Archive verwendet werden.

► `-z` bzw. `--gzip`

komprimiert bzw. dekomprimiert das gesamte Archiv durch `gzip`. Für das Anlegen von \*.tgz-Dateien ist diese Option sehr praktisch. Wenn dagegen tatsächlich Daten auf einem Streamer gespeichert werden, kann die Option gefährlich sein: Ein einziger Fehler auf dem Magnetband kann das gesamte Archiv unbrauchbar machen! Ohne Kompression werden dann zwar auch Dateien zerstört, im Regelfall ist der Schaden aber deutlich geringer. DAT-Streamer sind in der Lage, die zu verarbeitenden Daten selbst zu komprimieren. Das geht schneller, ist aber nicht so effektiv wie `gzip`.

## Beispiele

Im folgenden Beispiel archiviert `tar` alle Dateien aus dem Verzeichnis `Dokumente` und aus allen Unterverzeichnissen in der komprimierten Datei `meinarchiv.tgz`:

```
user$ tar -czf meinarchiv.tgz Dokumente
```

`tar -tzf` liefert ein Inhaltsverzeichnis des Archivs. Die Dateien innerhalb des Archivs sind willkürlich geordnet.

```
user$ tar -tzf meinarchiv.tgz | less
...
```

`tar -xzf` packt das Archiv aus und extrahiert alle enthaltenen Dateien:

```
user$ cd anderes-verzeichnis/
user$ tar -xzf meinarchiv.tgz
```

Das folgende Kommando kopiert alle Dateien aus `/verz1` nach `/verz2`. Der Vorteil gegenüber einem normalen `cp`-Kommando besteht darin, dass symbolische Links als solche kopiert werden (und nicht die Daten, auf die die Links verweisen). Die obigen zwei Kommando eignen sich besonders dazu, um ganze Dateibäume von einer Partition auf eine andere zu übertragen.

```
root# (cd /verz1 ; tar cf - .) | (cd /verz2 ; tar xvf -)
```

Das letzte Beispiel zeigt die Erzeugung inkrementeller Backups für das Verzeichnis `Bilder`. Beim ersten Backup wird eine eventuell schon vorhandene `Timestamp`-Datei gelöscht, um ein vollständiges Backup durchzuführen (z. B. immer sonntags nachts). Die weiteren Backups in der Nacht der folgenden Wochentage berücksichtigen dann nur die Änderungen, die seither erfolgt sind.

```
user$ rm backup.timestamps
user$ tar czvf backup-0.tgz -g backup.timestamps Bilder (So: vollst. Backup)
user$ tar czvf backup-1.tgz -g backup.timestamps Bilder (Mo: nur Änderungen)
user$ tar czvf backup-2.tgz -g backup.timestamps Bilder (Di: nur Änderungen)
```

### tasksel

tasksel installiert oder deinstalliert vordefinierte Paketgruppen. Das Kommando steht nur unter Debian- und Ubuntu-Distributionen zur Verfügung.

- ▶ `install gruppenname`  
installiert alle Pakete der angegebenen Paketgruppe.
- ▶ `--list-tasks`  
ermittelt eine Liste aller definierten Paketgruppen.
- ▶ `remove gruppenname`  
entfernt alle Pakete der angegebenen Paketgruppe. Vorsicht: Bei älteren tasksel-Versionen werden die Pakete ohne Rücksicht auf Abhängigkeiten zu anderen Paketen entfernt. Das führt dazu, dass oft wesentlich mehr Pakete als beabsichtigt deinstalliert werden.
- ▶ `--task-packages gruppenname`  
listet alle Pakete der angegebenen Paketgruppe auf.

### Beispiel

Die beiden folgenden Kommandos installieren unter Debian bzw. Ubuntu den Apache-Webserver sowie diverse oft benötigte Erweiterungen und Zusatzprogramme:

```
root# tasksel install web-server      (Debian)
root# tasksel install lamp-server     (Ubuntu)
```

### tee datei

tee dupliziert die Standardeingabe, zeigt eine Kopie an und speichert die andere in einer Datei. In der Praxis ist das dann sinnvoll, wenn die Ausgabe eines Kommandos auf dem Bildschirm beobachtet, aber gleichzeitig auch in einer Datei gespeichert werden soll. Eine einfache Umleitung mit `>` in eine Datei hätte zur Folge, dass auf dem Bildschirm nichts zu sehen ist.

### Beispiel

Das folgende Kommando zeigt das Inhaltsverzeichnis des aktuellen Verzeichnisses auf dem Bildschirm an und speichert es gleichzeitig in der Datei `inhalt`:

```
user$ ls -l | tee inhalt
```



```
telinit n
```

telinit wechselt in den angegebenen Runlevel. Bei Upstart-Systemen erzeugt das Kommando das Ereignis *runlevel n*.

```
telnet [optionen] host [port]
```

telnet erlaubt die interaktive Kommunikation mit einem Netzwerkdienst über das Telnet-Protokoll. Das Kommando eignet sich in erster Linie dazu, um elementare Netzwerkfunktionen zu testen. Da die Kommunikation unverschlüsselt erfolgt, ist telnet zu unsicher, um Kommandos auf externen Servern auszuführen. Für solche Aufgaben verwenden Sie besser ssh!

- ▶ `-4` bzw. `-6`  
akzeptiert ausschließlich IPv4- bzw. IPv6-Adressen.
- ▶ `-l user`  
verwendet den angegebenen Benutzernamen (anstelle des aktuellen Benutzernamens).

### Beispiel

Das folgende Beispiel zeigt die Kommunikation mit einem Mail-Server, der auf dem lokalen Rechner läuft:

```
user$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 kofler.info ESMTP Postfix (Ubuntu)
helo kofler.info
250 kofler.info
mail from:<kofler@kofler.info>
250 2.1.0 Ok
...
```

Oft sind Mail-Server so konfiguriert, dass nur die ersten Kommandos unverschlüsselt übertragen werden. Danach handeln Server und Client die Eckdaten der Verschlüsselung aus (STARTTLS-Verfahren). telnet muss an dieser Stelle passen, eignet sich dann also nur dazu, um grundsätzlich zu testen, ob der Mail-Server auf Port 25 aktiv ist.

**test** *ausdruck*

test wird in bash-Skripts zur Formulierung von Bedingungen verwendet und zumeist in if-Abfragen und Schleifen eingesetzt. Je nachdem, ob die Bedingung erfüllt ist, liefert es den Wahrheitswert 0 (wahr) oder 1 (falsch). Statt test kann auch die Kurzschreibweise [ *ausdruck* ] verwendet werden. Dabei ist es wichtig, dass Sie Leerzeichen vor und nach dem Ausdruck angeben!

Wenn test oder die Kurzschreibweise [ *ausdruck* ] als Bedingung in einer Verzweigung oder Schleife verwendet wird, muss die Bedingung mit einem Strichpunkt abgeschlossen werden, also z.B. if [ "\$1" = "abc" ]; then ...

if-Abfragen können manchmal durch die Formulierung [ "\$1" = "abc" ] && cmd ersetzt werden. In diesem Fall ist kein Strichpunkt erforderlich. Das Kommando wird nur ausgeführt, wenn die vorherige Bedingung erfüllt war.

**Zeichenketten**

[ <i>zk</i> ]	wahr, wenn die Zeichenkette nicht leer ist
[ -n <i>zk</i> ]	wahr, wenn die Zeichenkette nicht leer ist (wie [ <i>zk</i> ])
[ -z <i>zk</i> ]	wahr, wenn die Zeichenkette leer ist (0 Zeichen)
[ <i>zk1</i> = <i>zk2</i> ]	wahr, wenn die Zeichenketten übereinstimmen
[ <i>zk1</i> != <i>zk2</i> ]	wahr, wenn die Zeichenketten voneinander abweichen

Die Zeichenketten bzw. Variablen sollten in Hochkommata gestellt werden (z.B. [ "\$1" = "abc" ] oder [ "\$a" = "\$b" ]). Andernfalls kann es bei Zeichenketten mit mehreren Wörtern zu Fehlern kommen!

**Zahlen**

[ <i>z1</i> -eq <i>z2</i> ]	wahr, wenn die Zahlen gleich sind ( <i>equal</i> )
[ <i>z1</i> -ne <i>z2</i> ]	wahr, wenn die Zahlen ungleich sind ( <i>not equal</i> )
[ <i>z1</i> -gt <i>z2</i> ]	wahr, wenn <i>z1</i> größer <i>z2</i> ist ( <i>greater than</i> )
[ <i>z1</i> -ge <i>z2</i> ]	wahr, wenn <i>z1</i> größer gleich <i>z2</i> ist ( <i>greater equal</i> )
[ <i>z1</i> -lt <i>z2</i> ]	wahr, wenn <i>z1</i> kleiner <i>z2</i> ist ( <i>less than</i> )
[ <i>z1</i> -le <i>z2</i> ]	wahr, wenn <i>z1</i> kleiner gleich <i>z2</i> ist ( <i>less equal</i> )

**Dateien (auszugsweise)**

[ -d <i>dat</i> ]	wahr, wenn es sich um ein Verzeichnis handelt ( <i>directory</i> )
[ -e <i>dat</i> ]	wahr, wenn die Datei existiert ( <i>exist</i> )
[ -f <i>dat</i> ]	wahr, wenn es sich um eine einfache Datei (und nicht um ein Device, ein Verzeichnis ...) handelt ( <i>file</i> )
[ -L <i>dat</i> ]	wahr, wenn es sich um einen symbolischen Link handelt

[ -r <i>dat</i> ]	wahr, wenn die Datei gelesen werden darf ( <i>read</i> )
[ -s <i>dat</i> ]	wahr, wenn die Datei mindestens 1 Byte lang ist ( <i>size</i> )
[ -w <i>dat</i> ]	wahr, wenn die Datei verändert werden darf ( <i>write</i> )
[ -x <i>dat</i> ]	wahr, wenn die Datei ausgeführt werden darf ( <i>execute</i> )
[ <i>dat1</i> -ef <i>dat2</i> ]	wahr, wenn beide Dateien denselben I-Node haben ( <i>equal file</i> )
[ <i>dat1</i> -nt <i>dat2</i> ]	wahr, wenn Datei 1 neuer als Datei 2 ist ( <i>newer than</i> )

### Verknüpfte Bedingungen

[ ! <i>bed</i> ]	wahr, wenn die Bedingung nicht erfüllt ist
[ <i>bed1</i> -a <i>bed2</i> ]	wahr, wenn beide Bedingungen erfüllt sind ( <i>and</i> )
[ <i>bed1</i> -o <i>bed2</i> ]	wahr, wenn mindestens eine der Bedingungen erfüllt ist ( <i>or</i> )

### bash-Variante [[ *ausdruck* ]]

Während `test` und die Kurzschreibweise [ *ausdruck* ] in dieser Form in den meisten Shells funktionieren, steht in der `bash` auch eine Variante mit zwei eckigen Klammern zur Auswahl. Dort sind unter anderem die folgenden zusätzlichen Vergleiche bzw. Schreibweisen erlaubt:

[[ <i>zk</i> = <i>muster*</i> ]]	wahr, wenn die Zeichenkette mit <i>muster</i> beginnt
[[ <i>zk</i> == <i>muster*</i> ]]	wie oben
[[ <i>zk</i> = <i>regex</i> ]]	wahr, wenn die Zeichenkette dem regulärem Muster entspricht
[[ <i>bed1</i> && <i>bed2</i> ]]	wahr, wenn beide Bedingungen erfüllt sind ( <i>and</i> )
[[ <i>bed1</i>    <i>bed2</i> ]]	wahr, wenn mindestens eine der Bedingungen erfüllt ist ( <i>or</i> )

### Beispiel

Das folgende Mini-Script erfüllt keine sinnvolle Aufgabe, sondern demonstriert nur die `test`-Syntax in der Schreibweise [ *bedingung* ] && *kommando*. In der `for`-Schleife soll die Variable *i* die Werte von 1 bis 10 durchlaufen. *rest* enthält den Rest der ganzzahligen Division durch 2. Mit [ *\$rest* -eq 1 ] testet das Script, ob der Rest 1 beträgt, ob es sich bei *i* also um eine ungerade Zahl handelt. In diesem Fall werden die folgenden Kommandos mit `continue` übersprungen. `echo` gibt den Inhalt von *i* aus. Wenn *i* den Wert 6 erreicht, wird die Schleife mit `break` beendet. Somit gibt das Programm die Zahlen 2, 4 und 6 aus.

```
#!/bin/bash
for i in {1..10}; do
    rest=${i%2}
    [ $rest -eq 1 ] && continue
    echo $i
    [ $i -eq 6 ] && break
done
```

### **time** kommando

`time` führt das angegebene Kommando aus und gibt anschließend an, wie lange die Ausführung gedauert hat. `time` ersetzt nicht nur die Stoppuhr, sondern schlüsselt die Ausführungszeit auch in drei Komponenten auf:

- ▶ `real`  
gibt die tatsächliche Ausführungszeit aus.
- ▶ `user`  
gibt die CPU-Zeit an, die während der Ausführung des Programms außerhalb des Kernels gebraucht wurde.
- ▶ `sys`  
gibt die CPU-Zeit an, die während der Ausführung des Programms innerhalb des Kernels gebraucht wurde.

Die Summe aus `user` und `sys` gibt an, wie viel CPU-Aufwand die Ausführung des Kommandos verursacht hat. `real` ist aber oft (viel) größer als die Summe aus `user` und `sys`. Das kann zwei Gründe haben: Einerseits laufen parallel zum durch `time` gestarteten Kommando in der Regel eine Menge anderer Prozesse, die natürlich auch Rechenzeit kosten. Zum anderen kann es bei der Ausführung des Kommandos zu Wartezeiten kommen, etwa beim Lesen oder Schreiben von Dateien oder bei der Übertragung von Daten im Netzwerk.

`user` und `sys` beziehen sich jeweils auf *einen* CPU-Core. Wenn das ausgeführte Kommando z. B. mehrere Cores voll auslastet, kann die Summe aus `user` und `sys` größer als `real` sein.

### Beispiel

Das folgende Kommando liest den Inhalt eines Logical Volumes aus und komprimiert ihn mit dem Kommando `7zr` (Paket `p7zip`). Der Vorgang dauert auf einem Multi-Core-System im Leerlauf 24 Minuten. Würde nur ein Core zur Verfügung stehen, würde die Backup-Erstellung noch länger dauern (ca. 40 Minuten).

```
root# time dd if=/dev/vg830/lv3 bs=4M | 7zr a -si lv3.img.7z
real 24m53.424s
user 38m37.841s
sys 0m30.446s
```

```
timedatectl [optionen] [kommando]
```

Das Kommando `timedatectl` verändert bei Distributionen mit dem Init-System `Systemd` die Uhrzeit und die aktive Zeitzone. Die geänderten Einstellungen werden sofort wirksam; sie werden außerdem dauerhaft in `/etc/localtime` gespeichert.

- ▶ `list-timezones`  
liefert eine Liste aller bekannten Zeitzonen.
- ▶ `set-local-rtc 0|1`  
gibt an, ob die Uhr des Rechners die lokale Zeit oder die GMT-Zeit enthält.
- ▶ `settime datetime`  
stellt das Datum und die Uhrzeit neu ein. Dabei wird auch die Hardware-Uhr des Rechners entsprechend verändert. Die Zeitangabe muss in der Form `2016-12-31 23:59:59` erfolgen.
- ▶ `set-timezone name`  
stellt die gewünschte Zeitzone ein.
- ▶ `status`  
zeigt die Uhrzeit, die Zeitzone sowie diverse weitere Zeitdaten an, z. B. den nächsten Wechsel zwischen Sommer- und Winterzeit.

## Beispiel

Die folgenden Kommandos zeigen zuerst die aktuellen Zeitdaten und verändern dann die Zeitzone:

```
root# timedatectl status
    Local time: Mon 2016-10-10 13:06:24 CEST
    Universal time: Mon 2016-10-10 11:06:24 UTC
        Timezone: Europe/Brussels (CEST, +0200)
    ...
root# timedatectl set-timezone Europe/Vienna
```

```
timeout [optionen] zeitspanne kommando
```

`timeout` führt das angegebene Kommando aus und beendet es automatisch, wenn es länger als die angegebene Zeitspanne läuft. Die Zeitspanne wird als Fließkommazahl mit dem nachgestellten Buchstaben `s`, `m`, `h` oder `t` in Sekunden, Minuten, Stunden oder Tagen angegeben (z. B. `0.5h` für eine halbe Stunde).

## ► -k

beendet das zu lange laufende Programm durch ein KILL-Signal. Standardmäßig verwendet `timeout` ein TERM-Signal, das aber ignoriert werden kann.

**top** [q]

`top` zeigt alle fünf Sekunden die Liste aller laufenden Prozesse an, wobei die Prozesse nach ihrem Anteil an der Rechenzeit geordnet werden. Wenn beim Aufruf der optionale Parameter `q` angegeben wird, aktualisiert `top` die Liste ständig und beansprucht die gesamte freie Rechenzeit. `[q]` beendet das Programm.

Eine komfortablere Alternative zu `top` ist das Kommando `htop`, das bei den meisten Distributionen separat installiert werden muss. Es erlaubt ein horizontales und vertikales Scrollen in der Prozessliste. Wenn Sie sich nicht für den Speicherverbrauch und die CPU-Nutzung interessieren, sondern für die IO-Aktivität oder die Energieeffizienz der laufenden Prozesse, sollten Sie einen Blick auf die Beschreibung von `iotop` oder `powertop` werfen.

**touch** [optionen] dateien

`touch` verändert den zusammen mit der Datei gespeicherten Zeitpunkt der letzten Änderung. Wenn das Kommando ohne Optionen verwendet wird, wird als Änderungszeit die aktuelle Zeit gespeichert. Wenn die Datei noch nicht existiert, wird eine neue, 0 Byte lange Datei erzeugt.

► -r *datei*

verwendet die gespeicherte Änderungszeit der angegebenen Datei.

► -t *zeit*

speichert den angegebenen Zeitpunkt. Für die Zeitangabe müssen Sie das Format `[[CC]YY]MMDDhhmm[.ss]` verwenden, also z. B. als `201612311730` für den 31.12.2016 um 17:30 Uhr. Alternativ können Sie den Zeitpunkt auch als Zeichenkette mit der Option `-d` angeben. Dabei sind mehr besser lesbare Formate erlaubt, über die Sie sich mit `info touch` kundig machen können.

**Beispiel**

In diesem Beispiel wird `touch` dazu verwendet, die Änderungszeiten aller Dateien eines Verzeichnisses mit den Dateien eines zweiten Verzeichnisses abzugleichen. Das

ist z. B. dann sinnvoll, wenn Sie bei einem größeren `cp`-Kommando die Option `-a` vergessen haben und die kopierten Dateien nun alle das aktuelle Datum aufweisen.

```
user$ cd zielverzeichnis
user$ find . -type f -exec touch -r /quellverzeichnis/{ } {} \;
```

```
tr [optionen] zk1 [zk2] [ < quelle > ziel]
```

`tr` ersetzt in der angegebenen Quelldatei alle Zeichen der Zeichenkette 1 durch die entsprechenden Zeichen der Zeichenkette 2. Die beiden Zeichenketten sollten gleich lang sein. Zeichen, die in der ersten Zeichenkette nicht vorkommen, bleiben unverändert. Es ist nicht möglich, ein einzelnes Zeichen durch mehrere Zeichen zu ersetzen (etwa `ö` durch `"o"`) – dazu müssen Sie Kommandos wie `recode` oder `sed` verwenden.

► `-d`

löscht die in Zeichenkette 1 angegebenen Zeichen. Zeichenkette 2 braucht nicht angegeben zu werden.

### Beispiel

Im folgenden Beispiel ersetzt `tr` alle Klein- durch Großbuchstaben und zeigt das Ergebnis im Terminal (an der Standardausgabe) an:

```
user$ tr a-zäöü A-ZÄÖÜ < textdatei
```

```
tracert[6] [optionen] zieladresse
```

`tracert` liefert eine Liste aller Stationen, die auf dem Weg eines TCP/IP-Pakets vom lokalen Rechner zur Zieladresse liegen. Zu jeder Station wird die Gesamtlaufzeit angegeben (standardmäßig für drei Versuche). `tracert` funktioniert nur, wenn der UDP-Port 33434 auf dem Weg zur Zieladresse nicht durch eine Firewall blockiert wird. In diesem Fall liefert `tracert` statt der Angaben zur Zwischenstation nur drei Sterne.

► `-4` bzw. `-6`

nutzt ausschließlich das Protokoll IPv4 bzw. IPv6. Standardmäßig verwendet `tracert` automatisch das geeignete Protokoll. Wenn ein Hostname sowohl mit einer IPv4- als auch mit einer IPv6-Adresse verbunden ist, zieht `tracert` IPv4 vor. Für `tracert6` gilt automatisch `-6`.

► `-m n`

bestimmt die maximale Anzahl von Zwischenstationen (standardmäßig 30).

- ▶ `-n`  
zeigt bei den Zwischenstationen nur die IP-Nummer an (nicht den Hostnamen).
- ▶ `-p n`  
verwendet den angegebenen UDP-Port (standardmäßig 33434).
- ▶ `-q n`  
sendet  $n$  Pakete und misst für jedes Paket die Antwortzeiten (standardmäßig 3).

### Beispiel

Das folgende Kommando zeigt die Stationen zwischen meinem lokalen Rechner und meinem Webserver. Bei einigen Stationen werden die IP-Pakete über unterschiedliche Router geleitet, was auf eine redundante Netzwerkanbindung hindeutet.

```
root# traceroute kofler.info
traceroute to kofler.info (213.239.211.2), 30 hops max, 60 byte packets
 1 dslddevice.lan (10.0.0.138) 14.116 ms 13.376 ms 13.112 ms
 2 62.47.95.239 (62.47.95.239) 25.650 ms 27.489 ms 29.245 ms
 ...
 9 hos-tr3.ex3k16.rz6.hetzner.de (213.239.252.8) 48.315 ms
   hos-tr4.ex3k16.rz6.hetzner.de (213.239.252.136) 49.034 ms
   hos-tr2.ex3k16.rz6.hetzner.de (213.239.229.136) 51.231 ms
10 kofler.info (213.239.211.2) 52.069 ms 54.640 ms 55.757 ms
```

```
trap [kommando] signal
```

Das bash-Kommando `trap` führt den angegebenen Befehl aus, wenn die Shell das angegebene Signal empfängt. Wenn kein Kommando angegeben wird, ignoriert das Programm bzw. die bash das betreffende Signal. `trap -l` liefert eine Liste aller möglichen Signale und der ihnen zugeordneten Kennnummern. `trap` wird üblicherweise zur Fehlerabsicherung in Shell-Skripts eingesetzt.

### Beispiel

Das folgende Mini-Skript läuft in einer Endlosschleife, die sich nicht durch `[Strg]+[C]` unterbrechen lässt. Sie können das Skript aber z. B. durch `kill` beenden.

```
#!/bin/bash
trap 'echo "Strg+C wird ignoriert!"' SIGINT
while true
do
    sleep 1
done
```



```
tree [optionen] [startverzeichnis]
```

tree aus dem gleichnamigen Paket stellt die Hierarchie des Verzeichnisbaums ausgehend vom aktuellen bzw. vom angegebenen Startverzeichnis dar.

- ▶ -a  
gibt auch Dateien aus anstatt standardmäßig nur Verzeichnisse.
- ▶ -L n  
berücksichtigt nur die ersten n Hierarchieebenen.
- ▶ -x  
bleibt im aktuellen Dateisystem, berücksichtigt keine mount-Verzeichnisse oder Links in andere Dateisysteme.

```
truncate [optionen] dateien
```

truncate verkleinert oder vergrößert Dateien. Beim Verkleinern wird die Datei einfach abgeschnitten. Vorsicht: Dabei gehen Daten verloren! Beim Vergrößern werden der Datei entsprechend viele 0-Bytes hinzugefügt. Bei den meisten Dateisystemen ändert sich dadurch der Platzbedarf auf dem Datenträger nicht, weil die Datei als *Sparse File* erkannt wird. Wenn Sie eine Datei physikalisch mit Nullen füllen möchten, müssen Sie das Kommando `dd` einsetzen. Wenn an truncate der Name einer noch nicht vorhandenen Datei übergeben wird, erzeugt das Kommando die Datei in der gewünschten Größe.

- ▶ -s n  
gibt die gewünschte Größe in Byte an. Wenn der Zahl die Buchstaben K, M, G oder T folgen, multipliziert truncate die Zahl mit 1024, 1.048.576 etc. Alternativ stehen KB, MB, GB oder TB für 1000, 1.000.000 etc. 10K entspricht also 10.240 Byte, 10KB entspricht 10.000 Byte.

## Beispiel

Im folgenden Beispiel wird file zuerst mit 1 MByte Zufallsdaten gefüllt. Danach wird die Datei mit truncate auf 2 MByte vergrößert. `ls -l` zeigt tatsächlich eine Größe von 2 MByte, aber `dd` beweist, dass die Datei nur 1 MByte Platz auf der Festplatte oder SSD beansprucht. Der Platzbedarf steigt erst, wenn in die Datei später andere Daten als Nullen geschrieben werden.

```
user$ dd if=/dev/urandom of=file bs=1024 count=1024
user$ truncate -s 2M file
user$ ls -l file
-rw-r--r-- 1 kofler users 2097152 19. Nov 14:30 file
user$ du -h file
1,0M    file
```

### tty

tty zeigt den Device-Namen des aktiven Terminals an (für die Textkonsolen: /dev/tty1 bis tty6, für Shell-Fenster unter X /dev/pts/n).

### tune2fs [optionen] device

Mit tune2fs können Sie diverse Systemparameter eines ext2-/ext3-/ext4-Dateisystems verändern.

#### ► -c *n*

gibt an, nach wie vielen mount-Vorgängen die Partition beim Booten auf Fehler kontrolliert werden soll. 0 bedeutet, dass nie eine Überprüfung erfolgen soll.

#### ► -i *n*

gibt an, wie oft (in Tagen) die Partition beim Booten auf Fehler kontrolliert werden soll. 0 bedeutet, dass nie eine Überprüfung erfolgen soll.

#### ► -l

zeigt Informationen zum angegebenen Dateisystem an, ändert aber nichts.

#### ► -m *n*

gibt an, wie viel Prozent des Datenträgers für Daten von root reserviert werden sollen (bei mke2fs üblicherweise 5 Prozent).

#### ► -U *neue-uuid* bzw. -U random

gibt dem Dateisystem eine neue UUID.

### Beispiel

Das folgende Kommando bewirkt, dass das Dateisystem in der Partition /dev/sda1 nur noch einmal jährlich bzw. nach 200 mount-Vorgängen überprüft wird:

```
root# tune2fs -i 365 -c 200 /dev/sda1
```

Setze die maximale Mount-Anzahl auf 200  
Setze das Intervall zwischen Checks auf 31536000 Sekunden

### tvservice [optionen]

Unter Raspbian können Sie mit dem Kommando `tvservice` Informationen über den HDMI-Status des Raspberry Pi ermitteln und das HDMI-Signal aus- und wieder einschalten.

- ▶ `-e "group mode drive"` bzw. `--explicit "group mode drive"`  
schaltet den HDMI-Ausgang ein und aktiviert den angegebenen Modus. Anstelle von `group` geben Sie CEA oder DMT an, anstelle von `drive` das Schlüsselwort HDMI oder DVI. Die Liste der zur Auswahl stehenden Modi liefert `tvservice --modes CEA|DMT`.
- ▶ `-m CEA|DMT` bzw. `--modes=CEA|DMT`  
listet die vom Grafiksystem unterstützten Modi gemäß den Spezifikationen der *Consumer Electronics Association* (CEA) bzw. aus der Liste der von der VESA definierten *Display Monitor Timings* (DMT) auf. CEA-Modi sind auf TV-Geräten vorzuziehen, DMT-Modi auf Computer-Monitoren.
- ▶ `-o` bzw. `--off`  
schaltet den HDMI-Ausgang aus. Die meisten HDMI-Monitore aktivieren nun einen Energiesparmodus.
- ▶ `-p` bzw. `--preferred`  
aktiviert den HDMI-Ausgang und verwendet HDMI-Defaulteinstellungen. Leider funktioniert das Wiedereinschalten des Ausgangs auf dem Raspberry Pi nur bedingt: Der Monitor schaltet sich zwar wieder ein, das Bild bleibt aber schwarz. Abhilfe schafft nun ein zweimaliger Wechsel der aktiven Konsole mit dem Kommando `chvt`. Dabei müssen Sie zuerst eine Textkonsole und dann wieder die Grafikkonsole aktivieren (`chvt 1 && chvt 7`). Einfacher gelingt das Ein- und Ausschalten des HDMI-Ausgangs mit `vcgencmd display_power 0/1`.
- ▶ `-s` bzw. `--status`  
zeigt den aktuellen HDMI-Status aus (HDMI-Modus, Auflösung etc.).

### type kommando

Das bash-Kommando `type` ermittelt, ob es sich beim angegebenen Kommando um ein Shell-Kommando (beispielsweise `cd`), eine selbst definierte Funktion oder um eine

alias-Abkürzung handelt. Die Meldung *command is hashed* bedeutet, dass es sich um ein in dieser Sitzung bereits ausgeführtes Linux-Kommando handelt, dessen Pfadnamen sich die bash in einem Hash-Verzeichnis gemerkt hat.

### Beispiel

cd ist ein in die Shell integriertes Kommando:

```
root# type cd
cd is a shell builtin
```

**ufw** [optionen] kommando

ufw ist ein Ubuntu-spezifisches Kommando zur Firewall-Konfiguration. ufw ist unter Ubuntu und seinen Derivaten zwar standardmäßig installiert, aber nicht aktiviert. Wenn Sie ufw zur Firewall-Konfiguration verwenden möchten, müssen Sie die Firewall zuerst mit `ufw enable` aktivieren.

Zur Konfiguration der Firewall stehen unter anderem die folgenden Subkommandos zur Auswahl. Sie können mit der Option `--dry-run` kombiniert werden. In diesem Fall werden keine Änderungen durchgeführt. Die Option ist praktisch, um die Syntax komplexer Kommandos zu testen.

- ▶ `allow|deny [on interface] port/protocol/"app name"`  
erlaubt bzw. blockiert einen Port (z. B. 22), ein Protokoll (z. B. ssh) oder ein Programm, für das es eigene Regeldateien gibt (siehe `ufw app list`).  
  
Die Regel kann optional nur für eine bestimmte Netzwerkschnittstelle formuliert werden, also z. B. `on eth0`. man ufw zählt diverse weitere Optionen auf, um Regeln zu formulieren, die nur für eintreffende oder ausgehende Pakete bzw. nur für bestimmte Protokolle oder IP-Versionen gelten (z. B. für tcp, udp, ipv6).
- ▶ `app info "app name"`  
liefert detaillierte Informationen zu einer ufw-Regeldatei.
- ▶ `app list`  
liefert eine Liste aller Programme, für die es ufw-Regeldateien im Verzeichnis `/etc/uf2/applications.d` gibt.
- ▶ `default allow|deny|reject [incoming|outgoing|routed]`  
bestimmt das Defaultverhalten der Firewall. Ohne die Angabe von `incoming|outgoing|``routed` wird das Verhalten des Eingangsfilters verändert. Standardmäßig gilt

– sofern die Firewall überhaupt aktiviert ist – für den `incoming`-Filter das Verhalten `deny`, für den `outgoing`-Filter das Verhalten `allow`.

- ▶ `disable`  
deaktiviert die Firewall. Der gesamte Netzwerkverkehr kann ungehindert fließen.
- ▶ `enable`  
aktiviert die Firewall.
- ▶ `reload/reset`  
lädt alle Firewall-Regeln neu bzw. führt einen kompletten Neustart der Firewall durch.
- ▶ `status [verbose]`  
zeigt den Zustand der Firewall an.

### Beispiel

Die folgenden Kommandos erlauben zuerst den Dienst `ssh` und aktivieren dann die `ufw`-Firewall. Eine vorhandene SSH-Verbindung sollte dadurch nicht unterbrochen werden. Alle anderen Server-Dienste werden aber von nun an blockiert.

```
root# ufw allow ssh
root# ufw enable
root# ufw status verbose
Status: Aktiv
Protokollierung: on (low)
Voreinstellung: deny (eingehend),
                allow (abgehend),
                disabled (gesendet)
Neue Profile: skip
Zu           Aktion      Von
--           -
22           ALLOW IN   Anywhere
22 (v6)      ALLOW IN   Anywhere (v6)
```

### **ulimit** option grenzwert

`ulimit` begrenzt die Systemressourcen, die von der Shell und den aus ihr gestarteten Prozessen in Anspruch genommen werden. Größenangaben erfolgen generell in `kByte`. `ulimit` wird oft in `/etc/profile` voreingestellt.

- ▶ `-c speicher`  
beschränkt die Größe von Coredumps, also des Speicherabbaus, das bei einem Programmabsturz automatisch auf der Festplatte gespeichert wird.
- ▶ `-d speicher`  
beschränkt den Speicher für das Datensegment von Prozessen.
- ▶ `-f dateigröße`  
verhindert die Erzeugung von Dateien, die größer als der angegebene Grenzwert sind. Funktioniert nicht mit allen Dateisystemen.
- ▶ `-s speicher`  
beschränkt den Stack-Speicher.

### `umask` [maske]

Das bash-Kommando `umask` steuert, mit welchen Zugriffsrechten neue Dateien angelegt werden. `umask` merkt sich hierfür einen oktalen Zahlenwert, der von den Defaultzugriffsrechten neuer Dateien bzw. Verzeichnisse abgezogen wird.

Linux sieht eigentlich vor, dass neue Dateien die Zugriffsbits `rw-rw-rw` (oktal 666) bekommen. Neue Verzeichnisse und Programmdateien, die von einem Compiler erzeugt werden, bekommen automatisch die Zugriffsbits `rw-rw-rw` (777). Für die praktische Arbeit sind diese Grundeinstellungen zu großzügig. Deswegen sehen alle Linux-Shells die sogenannte `umask`-Einstellung vor. Dabei handelt es sich um einen Zahlenwert, der die Zugriffsbits angibt, die von den Standardzugriffsbits *abgezogen* werden.

Die `umask`-Einstellung ist systemweit vorgegeben und wird je nach Distribution in `/etc/profile` oder in `/etc/bashrc` eingestellt. Oft gelten für `root` und für alle anderen Benutzer unterschiedliche Werte. Bei Ubuntu ist der `umask`-Wert durch das PAM-System festgelegt; der Defaultwert ist in `/etc/login.defs` gespeichert.

### Beispiele

`umask` ohne weitere Parameter zeigt die aktuelle Einstellung an:

```
michael$ umask (Debian, openSUSE, Ubuntu)
0022
michael$ umask (Fedora, RHEL)
0002
```

Das zweite Beispiel zeigt, welche Auswirkungen es auf neue Dateien und Verzeichnisse hat, wenn Sie den `umask`-Wert auf 027 ändern. Neue Dateien erhalten so die Zugriffsrechte  $666 - 026 = 640 = \text{rw-r-----}$ , neue Verzeichnisse  $777 - 027 = 750 = \text{rwxr-x---}$ .

```
michael$ umask 27
michael$ touch neue-datei
michael$ mkdir neues-verzeichnis
michael$ ls -ld neu*
-rw-r----- ... michael michael ... neue-datei
drwxr-x--- ... michael michael ... neues-verzeichnis
```

```
umount device
umount verzeichnis
```

`umount` entfernt ein Dateisystem aus dem Linux-Verzeichnisbaum. Die Angabe des Dateisystems erfolgt entweder durch den Device-Namen des Datenträgers oder durch die Angabe des Verzeichnisses, bei dem das Dateisystem in den Verzeichnisbaum eingebunden ist. Das Kommando kann nur von `root` ausgeführt werden. Es führt zu einer Fehlermeldung, wenn es auf dem Dateisystem noch geöffnete Dateien gibt.

► `-f`

erzwingt den `umount`-Vorgang (zweckmäßig bei nicht mehr erreichbaren NFS-Verzeichnissen).

```
unalias abkürzung
```

`unalias` löscht die angegebene Abkürzung. Wenn das Kommando mit der Option `-a` aufgerufen wird, löscht es alle bekannten Abkürzungen. Der Umgang mit Abkürzungen wird bei der Beschreibung des `alias`-Kommandos erläutert.

```
uname [optionen]
```

`uname` zeigt den Namen des Betriebssystems an (also Linux). Durch die Angabe von Optionen können auch andere Informationen angezeigt werden.

► `-a`

zeigt alle verfügbaren Informationen an, nämlich das Betriebssystem, die Versionsnummer, Datum und Uhrzeit sowie den Prozessor.

► `-m`

liefert nur die CPU-Plattform (z. B. `i686` oder `x86_64` bei 64-Bit-Systemen).

► `-r`

liefert nur die Kernelversion.

### Beispiel

Auf diesem Rechner läuft ein 64-Bit-Kernel, der im November 2015 kompiliert wurde:

```
root# uname -a
Linux localhost.localdomain 4.2.6-301.fc23.x86_64
#1 SMP Fri Nov 20 22:22:41 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
```

**uncompress** datei

`uncompress` dekomprimiert eine mit `compress` komprimierte Datei. Dabei wird die Dateikennung `.Z` automatisch entfernt. `uncompress` ist ein Link auf `compress`, wobei automatisch die Option `-d` aktiviert wird.

**unexpand** textdatei > ergebnis

`unexpand` ersetzt mehrfache Leerzeichen in der angegebenen Textdatei durch Tabulatorzeichen und schreibt das Ergebnis in die Standardausgabe. Das Kommando ist das Gegenstück zu `expand`, das Tabulatorzeichen durch Leerzeichen ersetzt.

► `-a`

ersetzt alle Leerzeichen, nicht nur die am Beginn einer Zeile.

► `-t n`

gibt die Anzahl der Zeichen je Tabulatorposition an (standardmäßig 8).

**uniq** datei

`uniq` gibt die Zeilen einer Textdatei auf der Standardausgabe aus, wobei unmittelbar aufeinanderfolgende gleichlautende Zeilen eliminiert werden. Bei vorsortierten Dateien eliminiert `uniq` alle mehrfach auftretenden Zeilen.

### Beispiel

`sort` sortiert die Datei `test`, und `uniq` eliminiert doppelte Zeilen und speichert das Resultat in `test1`:

```
user$ sort test | uniq > test1
```



```
unset variable
```

Das bash-Kommando `unset` löscht die angegebene Variable.

```
until bedingung; do  
    kommandos  
done
```

`until` bildet Schleifen in bash-Skripts. Die Schleife wird so lange ausgeführt, wie die angegebene Bedingung erfüllt ist. Das Schleifenkriterium ist der Rückgabewert des Kommandos, das als Bedingung angegeben wird. Vergleiche und Tests werden mit dem Kommando `test` oder dessen Kurzform in eckigen Klammern durchgeführt.

### Beispiel

Das Beispiel zeigt die Formulierung einer simplen Schleife mit `until`:

```
user$ i=1; until [ $i -gt 3 ]; do echo $i; i=$((i+1)); done  
1  
2  
3
```

```
unxz [optionen] dateien
```

`unxz` aus dem Paket `xz-utils` dekomprimiert die zuvor mit `xz` komprimierten Dateien. Dabei wird die Dateiendung `.xz` entfernt.

```
unzip [optionen] archiv.zip [datei1 datei2]
```

Das Kommando extrahiert Dateien aus einem vor allem in der Windows-Welt sehr verbreiteten ZIP-Archiv. Wenn die zu extrahierenden Dateien nicht explizit aufgeführt werden, packt `unzip` alle Dateien des Archivs aus. Wenn Sie unter Linux selbst ZIP-Archive bilden möchten, verwenden Sie hierfür `zip`.

### Beispiel

Das folgende Kommando extrahiert die Datei `druck.pdf` aus `archiv.zip`. Sollte es in diesem Archiv noch weitere Dateien geben, werden diese nicht angerührt.

```
root# unzip archiv.zip druck.pdf  
Archive:  ../archiv.zip  
  inflating: druck.pdf
```

**update-alternatives** [optionen] kommando

update-alternatives verwaltet die Links im Verzeichnis `/etc/alternatives`. Diese Links bestimmen die aktive Version mehrerer parallel installierter Programme mit derselben Funktion (z. B. Editoren oder Java-Umgebungen). Das Kommando wird in der Regel von den (De-)Installations-Skripts der betreffenden Pakete ausgeführt, kann aber natürlich auch interaktiv verwendet werden. Unter Red Hat bzw. Fedora ist das Kommando auch unter dem Namen alternatives verfügbar.

- ▶ `--auto name`  
aktiviert den Automatikmodus für das Kommando. Damit ist automatisch das Kommando aktiv, das den höchsten Prioritätswert hat.
- ▶ `--config name`  
zeigt die zur Auswahl stehenden Alternativen zum angegebenen Kommando an. Anschließend geben Sie interaktiv an, welche Alternative in Zukunft aktiv sein soll. update-alternatives passt die Links anschließend entsprechend an und wechselt für das Kommando in den manuellen Modus.
- ▶ `--display name`  
liefert eine Liste aller installierten Alternativen zum angegebenen Kommando.
- ▶ `--install ...` bzw. `--remove ...`  
installiert bzw. entfernt eine Alternative für ein Kommando. Diese Kommandos kommen in der Regel nur im (De-)Installations-Skript eines Pakets zur Anwendung.
- ▶ `--set name kommandopfad`  
richtet das mit *kommandopfad* angegebene Kommando als Standardprogramm für *name* ein.

**Beispiel**

Das folgende Kommando definiert `/usr/bin/jmcs` als Standardeditor:

```
root# update-alternatives --set editor /usr/bin/jmcs
```

**update-grub**

Dieses Skript steht nur unter Ubuntu und Debian zur Verfügung. Seine Wirkung hängt von der installierten GRUB-Version ab:

► **GRUB 0.97:** `update-grub`

wertet die in Kommentaren versteckten Einstellungen in `/boot/grub/menu.lst` aus und aktualisiert dann die GRUB-Menüeinträge in `menu.lst`.

► **GRUB 2:** `update-grub`

führt die Konfigurations-Skripts `/etc/grub.d/*` aus und erstellt die neue GRUB-Menüdatei `/boot/grub/grub.cfg`. Wenn Sie nicht unter Debian oder Ubuntu arbeiten, führen Sie statt `update-grub` das Kommando `grub2-mkconfig -o /boot/grub2/grub.cfg` aus.

### **update-initramfs** [optionen]

`update-initramfs` ist bei Debian und Ubuntu für das Erzeugen, Aktualisieren oder Löschen von Initrd-Dateien zuständig. Die Initrd-Dateien enthalten Kernelmodule, die während des Systemstarts durch GRUB geladen werden. Bei Fedora, Red Hat und SUSE verwenden Sie statt `update-initramfs` das Kommando `dracut`.

Intern greift `update-initramfs` auf das Script `mkinitramfs` zurück, dessen direkte Verwendung aber nicht empfohlen wird. Die Konfiguration der Initrd-Dateien erfolgt durch die Dateien des Verzeichnisses `/etc/initramfs-tools`.

► `-c`

erzeugt eine neue Initrd-Datei für die mit `-k` angegebene Kernelversion.

► `-d`

löscht die Initrd-Datei für die mit `-k` angegebene Kernelversion.

► `-k versionsname`

gibt die zu bearbeitende Kernelversion an. `-k all` bewirkt, dass `update-initramfs` die Initrd-Dateien aller installierten Kernelversionen bearbeitet.

► `-u`

aktualisiert die Initrd-Datei der aktuellsten Kernelversion bzw. der mit `-k` angegebenen Kernelversion.

### Beispiel

Das folgende Kommando erzeugt eine neue Initrd-Datei für die Kernelversion 3.12.3. Die Kerneldatei hat den Dateinamen `/boot/vmlinuz-3.12.3-45-generic`, und die resultierende Initrd-Datei heißt `/boot/initrd.img-3.12.3-45-generic`.

```
root# update-initramfs -c -k 3.12.3-45-generic
```

**updatedb**

updatedb erstellt ein Indexverzeichnis für das locate-Kommando. Der Index enthält eine Liste aller im gesamten Dateisystem enthaltenen Dateien. Das Kommando wird üblicherweise einmal täglich automatisch durch einen Cron-Job ausgeführt. Die Ausführung erfordert root-Rechte. Je nach Distribution wird die Dateidatenbank im Verzeichnis `/var/lib`, `/var/lib/slocate` oder `/var/lib/mlocate` gespeichert.

**useradd** [optionen] name

useradd richtet einen neuen Benutzer ein.

- ▶ `-c "vollständiger Name"`  
gibt den vollständigen Namen des neuen Benutzers an.
- ▶ `-g gruppe`  
gibt die Hauptgruppe (primäre Gruppe) des Benutzers an.
- ▶ `-G gruppeA,gruppeB,gruppeC`  
bestimmt alle Zusatzgruppen (Supplementary Groups) des Benutzers.
- ▶ `-m`  
falls noch kein Heimatverzeichnis existiert (`/home/name`), wird es erzeugt. Alle Dateien aus `/etc/skel` werden dorthin kopiert. Normalerweise gilt `-m` aufgrund der Voreinstellungen in `/etc/login.defs` standardmäßig und muss nicht angegeben werden. In solchen Fällen kann die Erzeugung des Heimatverzeichnisses durch `-M` verhindert werden.
- ▶ `-u n`  
weist dem Benutzer die angegebene UID-Nummer zu (User Identification), sofern diese noch verfügbar ist.

**Beispiel**

Die folgenden Kommandos richten den neuen Benutzer Gerald Gersin mit dem Login-Namen `gersin` ein, definieren ein initiales Passwort und zwingen den Benutzer, sofort beim ersten Login und in der Folge alle 100 Tage ein neues Passwort einzustellen:

```
root# useradd gersin -c "Gerald Gersin"
root# passwd gersin
Geben Sie ein neues Passwort ein: ******
```

Geben Sie das neue Passwort erneut ein: \*\*\*\*\*  
 root# `chage -d 0 -M 100 gersin`

**userdel** name

userdel löscht den angegebenen Benutzer-Account.

► `-r`

löscht auch das gesamte Heimatverzeichnis sowie die Mail-Inbox des Benutzers.

**usermod** [optionen] name

usermod verändert diverse Eigenschaften des Benutzer-Accounts, z. B. das Heimatverzeichnis, die Gruppenzugehörigkeit, die Standard-Shell oder die UID. Die meisten Optionen sind mit denen von useradd identisch. Änderungen am Benutzer-Account werden erst nach einem neuerlichen Login wirksam.

► `-a -G gruppe`

fügt den Benutzer der angegebenen Gruppe hinzu.

► `-L`

blockiert den Account vorübergehend. Dazu wird in `/etc/shadow` vor dem Hash-code des Passworts das Zeichen `»!»` gestellt, weswegen ein Login nicht mehr möglich ist.

► `-U`

gibt einen durch `-L` blockierten Account wieder frei.

### Beispiel

Das folgende Kommando fügt den Benutzer gersin der Gruppe docuteam hinzu:

root# `usermod -a -G docuteam gersin`

**vcgencmd** kommando

Mit dem spärlich dokumentierten Kommando `vcgencmd` können Sie unter Raspbian Eckdaten des Geräts bzw. seiner CPU auslesen und einige wenige Parameter auch verändern.

- ▶ `codec_enabled H264|MPG2|WVC1|MPG4|MJPEG|WMV9`  
gibt an, ob der betreffende Hardware-Codec genutzt werden kann. Die MPEG2- und VC1-Codecs erfordern einen Lizenzschlüssel, der in /boot/config.txt eingetragen werden muss.
- ▶ `commands`  
listet alle `vcgencmd`-Kommandos auf.
- ▶ `display_power 0|1`  
schaltet den HDMI-Ausgang aus bzw. wieder ein.
- ▶ `get_config parametername |int|str`  
liefert den Zustand des angegebenen Parameters bzw. aller Integer- oder String-Parameter. Die Parameter können in der Datei `/boot/config.txt` eingestellt werden.
- ▶ `measure_clock arm|core|hdmi|uart`  
liefert die CPU-Frequenz, die Frequenz der Grafik-Cores sowie die Frequenz diverser weiterer Komponenten.
- ▶ `measure_temp`  
liefert die Temperatur der CPU.
- ▶ `measure_volts core|sdram_c|sdram_i|sdram_p`  
gibt an, mit welcher Spannung die Grafik-Cores und der Speicher versorgt werden.
- ▶ `version`  
liefert die Firmware-Version.

### Beispiel

Die folgenden Kommandos zeigen den Betriebszustand eines Raspberry Pi, der gerade im Leerlauf ist:

```
pi$ vcgencmd measure_clock arm
frequency(45)=600000000
pi$ vcgencmd measure_temp
temp=44.4'C
```

```
vgchange [optionen] [vgname]
```

Das LVM-Kommando `vgchange` verändert die Attribute einer Volume Group (VG). Die wichtigste Anwendung besteht darin, VGs zu aktivieren bzw. zu deaktivieren.

► `-a y|n`

aktiviert (y) bzw. deaktiviert (n) alle VGs bzw. die angegebene VG.

```
vgcreate [optionen] vgname pvname1 [pvname2 ...]
```

`vgcreate` erzeugt eine neue VG aus einem oder mehreren Physical Volumes (PVs).

### Beispiel

Im folgenden Beispiel markiert zuerst `pvcreate` die jeweils erste Partition der Festplatten oder SSDs `/dev/sdb` und `/dev/sdc1` als Physical Volumes. `vgcreate` bildet daraus die Volume Group `vg1`. Der Befehl `lvcreate` reserviert nun 100 GByte für das Logical Volume `lv1`. `mkfs.ext4` richtet darin ein Dateisystem ein, und `mount` bindet dieses am gerade erzeugten Verzeichnis `/mnt/lv1` in den Verzeichnisbaum ein.

```
root# pvcreate /dev/sdb1
root# pvcreate /dev/sdc1
root# vgcreate vg1 /dev/sdb1 /dev/sdc1
root# lvcreate -L 100G -n lv1 vg1
root# mkfs.ext4 /dev/mapper/vg1-lv1
root# mkdir /mnt/lv1
root# mount /dev/mapper/vg1-lv1 /mnt/lv1
```

```
vgdisplay vgname
```

`vgdisplay` zeigt Detailinformationen zur angegebenen VG an.

```
vgextend vgname pvname
```

`vgextend` fügt ein PV zu einer VG hinzu.

### Beispiel

Die folgenden Kommandos definieren die Partition `/dev/sdc2` als neues Physical Volume und fügen dieses dann der vorhandenen, zu klein gewordenen Volume Group `myvg1` hinzu:

```
root# pvccreate /dev/sdc2
Physical volume "/dev/sdc2" successfully created
root# vgextend myvg1 /dev/sdc2
Volume group "myvg1" successfully extended
root# vdisplay myvg1
...
VG Size                180,64 GB
Alloc PE / Size        6402 / 22,50 GB
Free PE / Size         41324 / 158,14 GB
...
```

**vgmerge** vname1 vname2

vgmerge fügt VG2 zu VG1 hinzu. Nach der erfolgreichen Ausführung des Kommandos gibt es also nur noch VG1. Sie besteht aus allen PVs, die bisher den Speicherpool für VG1 und VG2 bildeten.

**vgreduce** [optionen] vname [pvname1 pvname2 ...]

vgreduce entfernt die angegebenen PVs aus dem Speicherpool der VG. Das funktioniert nur, wenn die PVs ungenutzt sind. Mit der Option -a werden alle inaktiven PVs entfernt.

**vgrename** oldvname newvname

vgrename gibt einer VG einen neuen Namen. Statt oldvname kann auch die UUID der VG angegeben werden.

**vgscan**

vgscan listet alle VGs auf.

**virsh** [[-c connection] kommando]

Mit dem Kommando virsh administrieren Sie virtuelle KVM- und Xen-Maschinen. virsh kann auf zwei Arten verwendet werden: entweder zur direkten Ausführung eines virsh-Befehls oder interaktiv als Shell. Bei der ersten Variante können Sie mit der Option -c eine Verbindungszeichenkette angeben:



```
root# virsh -c qemu:///session list --all
...
```

Im Folgenden sind die wichtigsten virsh-Kommandos kurz beschrieben. Sollten Sie weitere Details in der *man*-Seite nachlesen, beachten Sie bitte, dass virtuelle Maschinen in der virsh-Nomenklatur *Domänen* heißen.

- ▶ `attach-device name device.xml [--persistent]`  
fügt einer virtuellen Maschine eine zusätzliche Hardware-Komponente hinzu (z. B. ein USB-Gerät), die im libvirt-XML-Format beschrieben ist. Die Option `--persistent` bewirkt, dass das Gerät bleibend mit der virtuellen Maschine verbunden wird und in der XML-Datei der virtuellen Maschine gespeichert wird.
- ▶ `attach-disk name source target`  
fügt einer virtuellen Maschine einen Datenträger hinzu. Dabei ist *source* der Device-Name auf dem Host-System, *target* der Device-Name im Gast. Das Kommando kann durch zahlreiche Optionen ergänzt werden, die den Treiber (`--driver`), das Caching-Verfahren (`--cache`) etc. angeben. Mit der Option `--persistent` wird der Datenträger bleibend mit der virtuellen Maschine verbunden.
- ▶ `attach-interface name type source`  
fügt einer virtuellen Maschine eine Netzwerkschnittstelle hinzu. *type* gibt den Typ der Schnittstelle an, z. B. *network* oder *bridge*. *source* gibt den Schnittstellenamen auf dem Hostrechner an (z. B. *br0*). Die Details der Schnittstelle können durch weitere Optionen eingestellt werden (`--mac`, `--model` etc.).
- ▶ `autostart [--disable] name`  
gibt an, dass die virtuelle Maschine während des Bootprozesses des Hostrechners automatisch gestartet werden soll. Mit der Option `--disable` wird der automatische Start wieder abgestellt.
- ▶ `connect qemu:///session`  
stellt eine Benutzerverbindung zur libvirtd-Instanz des aktuellen Benutzers her. Auf diese Weise können eigene virtuelle Maschinen verwaltet werden.
- ▶ `connect qemu:///system`  
stellt eine Verbindung zur Systeminstanz von libvirtd her. Wenn Sie virsh mit root-Rechten ausführen, stellt virsh diese Verbindung automatisch her.
- ▶ `connect qemu+ssh://user@hostname/system`  
stellt eine Verbindung zur libvirtd-Instanz eines anderen Rechners (*hostname*) her. Die Kommunikation erfolgt über einen SSH-Tunnel.

- ▶ `console name`

ermöglicht die Bedienung der angegebenen virtuellen Maschine direkt in der Konsole. Das setzt voraus, dass in der virtuellen Maschine ein `getty`-Prozess für die serielle Schnittstelle `/dev/ttyS0` läuft. Um die Verbindung zu beenden, drücken Sie `[Strg]+[J]`.
- ▶ `define xmldatei`

richtet eine neue virtuelle Maschine ein, deren Eckdaten in der angegebenen XML-Datei zusammengefasst sind. Vorsicht: Wenn bereits eine gleichnamige virtuelle Maschine existiert (gemäß dem `<name>`-Element in der XML-Datei), wird deren Definition überschrieben!
- ▶ `destroy name`

beendet die virtuelle Maschine sofort. Das ist so, als würden Sie bei Ihrem Rechner das Stromkabel ausstecken, und es kann dieselben Folgen haben (also ein zerstörtes Dateisystem etc.)!
- ▶ `detach-device name device.xml`

entfernt die durch eine XML-Datei beschriebene Hardware-Komponente von der virtuellen Maschine.
- ▶ `detach-disk name target`

entfernt den Datenträger von der virtuellen Maschine. `target` gibt den Device-Namen im Gast an.
- ▶ `detach-interface name type --mac=xxx`

löst eine Netzwerkschnittstelle von der virtuellen Maschine.
- ▶ `domstatus name` und `dominfo name`

liefern Informationen zu einer virtuellen Maschine.
- ▶ `edit name`

lädt die XML-Datei zur Beschreibung der virtuellen Maschine in einen Editor, wobei die Umgebungsvariable `$EDITOR` beachtet wird.
- ▶ `help` bzw. `help kommando`

liefert eine Liste aller Kommandos bzw. die Syntaxbeschreibung eines bestimmten Kommandos.

- `list [--inactive oder --all]`

listet alle laufenden virtuellen Maschinen auf. Wenn Sie nur die gerade nicht aktiven oder aber alle Maschinen auflisten möchten, geben Sie die Optionen `--inactive` oder `--all` an.

- `managedsave name`  
`managedsave-remove name`

speichert den Zustand der virtuellen Maschine (also den Inhalt des RAMs, die CPU-Register etc.) in einer Datei im Verzeichnis `/var/lib/libvirt/save/` und stoppt dann die Ausführung der Maschine. Zur Reaktivierung der virtuellen Maschine verwenden Sie einfach `start`. Die Zustandsdatei wird dann automatisch gelöscht. Wenn Sie den gespeicherten Zustand verwerfen und die virtuelle Maschine von der Festplatte neu starten möchten, löschen Sie die Zustandsdatei mit `managedsave-remove`.

- `net-create xmlfile`  
`net-start netname`  
`net-destroy netname`  
`net-undefine netname`  
`net-list`

helfen bei der Verwaltung virtueller Netzwerke. Dabei handelt es sich um private Netzwerkbereiche, die via NAT mit dem Hostsystem verbunden werden können – beispielsweise das default-Netzwerk der libvirt-Werkzeuge.

- `pool-define xmlfile`  
`pool-define-as poolname type --target path`  
`pool-start poolname`  
`pool-auto-start xmlfile`  
`pool-destroy poolname`  
`pool-delete poolname`  
`pool-list`  
`pool-info poolname`

helfen bei der Administration von libvirt-Speicher-Pools. `pool-define` erzeugt einen neuen Pool, dessen Eigenschaften in einer XML-Datei beschrieben sind. `pool-define-as` erzeugt ebenfalls einen neuen Pool, wobei die Eckdaten direkt als Parameter übergeben werden. Erlaubte Pool-Typen sind unter anderem `dir` (ein lokales Verzeichnis), `netfs` (ein Netzwerkverzeichnis), `logical` (eine Volume Group), `disk` (eine Festplatte) oder `iscsi` (ein iSCSI-Server).

Der neue Pool muss anschließend mit `pool-start` gestartet werden. Wenn der Pool in Zukunft automatisch gestartet werden soll, müssen Sie außerdem `pool-autostart` ausführen.

Etwas verwirrend sind die Kommandos zum Löschen eines Pools: Ein Pool muss vor dem Löschen mit `pool-destroy` deaktiviert werden. Obwohl das Kommando Schlimmes vermuten lässt, wird der Pool dadurch lediglich gestoppt und kann später mit `pool-start` wieder gestartet werden. Erst `pool-delete` löscht den Pool. `pool-delete` setzt voraus, dass zuerst alle Volumes des Pools gelöscht wurden. Es kann also nur ein leerer Pool gelöscht werden.

- ▶ `qemu-monitor-command --hmp name 'kommando'`  
führt das angegebene QEMU-Monitor-Kommando für die durch *name* angegebene virtuelle Maschine aus. Die Option `--hmp` ist erforderlich, weil das Kommando andernfalls im JSON-Format angegeben werden muss.
- ▶ `restore datei`  
aktiviert eine mit `save` gespeicherte virtuelle Maschine wieder. Die Zustandsdatei kann anschließend gelöscht werden.
- ▶ `save name datei`  
speichert den Zustand der virtuellen Maschine (also im Wesentlichen den Inhalt des RAMs) in einer Datei und stoppt dann die Ausführung der Maschine. Zur Reaktivierung der virtuellen Maschine verwenden Sie `restore`.
- ▶ `schedinfo [optionen] name`  
zeigt die Scheduler-Parameter der virtuellen Maschine an bzw. verändert diese (`--set parameter=wert`). Mit diesen Parametern kann gesteuert werden, wie viele Hardware-Ressourcen eine virtuelle Maschine nutzen darf. Die Ressourcensteuerung erfordert die Aktivierung der `cgroups`-Funktionen auf dem Host-System.
- ▶ `shutdown/reboot name`  
fährt die virtuelle Maschine herunter bzw. startet sie neu. Die virtuelle Maschine erhält via ACPI ein Shutdown-Signal. In der virtuellen Maschine muss der ACPI-Dämon `acpid` installiert sein, damit das Signal auch verarbeitet wird.
- ▶ `snapshot-create vmname [xml-datei [--redefine]]`  
`snapshot-create-as vmname snapshotname [beschreibung]`  
`snapshot-list vmname`  
`snapshot-delete vmname sname`  
`snapshot-revert vmname sname [--running]`  
erzeugt einen Snapshot einer laufenden virtuellen Maschine, listet alle Snapshots auf, löscht einen Snapshot wieder bzw. wendet den Inhalt eines Snapshots auf die Image-Datei an. Die Snapshot-Funktion kann nur für virtuelle Maschinen verwendet werden, die QCOW2-Image-Dateien nutzen.

► `start name`

startet die angegebene virtuelle Maschine. Wenn Sie mit der Maschine im Grafikmodus kommunizieren möchten, verwenden Sie dazu entweder einen VNC-Client (die Verbindungsdaten ermittelt das `virsh`-Kommando `vncdisplay`, siehe unten) oder das Programm `virt-viewer`.

► `suspend/resume name`

stoppt die angegebene virtuelle Maschine vorübergehend bzw. setzt die Ausführung wieder fort. Die gestoppte virtuelle Maschine beansprucht jedoch weiterhin RAM! Es wird also nur die virtuelle CPU angehalten.

► `ttyconsole name`

gibt an, über welches Device des Hostcomputers die serielle Schnittstelle des Gastsystems zugänglich ist (z. B. `/dev/pts/5`).

► `undefine name`

löscht die XML-Datei, die die virtuelle Maschine beschreibt. Die Image-Datei mit der virtuellen Festplatte bleibt erhalten. `undefine` kann erst ausgeführt werden, nachdem alle Snapshots der virtuellen Maschine gelöscht wurden.

► `vcpuinfo name`

liefert Informationen zu den (virtuellen) CPUs, die der virtuellen Maschine zur Verfügung stehen, sowie Angaben zur bisher beanspruchten CPU-Zeit. Mit dem Kommando kann auch das CPU-Pinning überprüft werden.

► `vcupin name gast-cpu-nr host-cpu-list`

verknüpft die physikalischen CPU-Cores des Hostsystems mit den virtuellen CPUs des Gasts. `vcupin vm1 0 2` ordnet der ersten virtuellen CPU der virtuellen Maschine `vm1` den dritten CPU-Core des Hosts `fix` zu.

► `vol-create xmlfile`

`vol-create-as poolname newvolname size`

`vol-delete volname`

`vol-list`

`vol-info volname`

helfen bei der Administration von Datenträgern in Speicher-Pools. `vol-create` erzeugt einen neuen Datenträger, dessen Eigenschaften in der angegebenen XML-Datei beschrieben sind. `vol-create-as` erzeugt einen neuen Datenträger in der gewünschten Größe, wobei die Suffixe `k`, `M`, `G` und `T` für `kByte`, `MByte`, `GByte` und `TByte` zulässig sind. Wenn Sie eine Image Disk erzeugen, können Sie mit `--format raw/qcow2/qed` das gewünschte Format angeben (standardmäßig `raw`).

--allocation *size* bestimmt, wie viel des Speichers im Voraus alloziert werden soll; bei RAW-Volumes wird immer der gesamte Speicher reserviert.

Wenn es mehrere Speicher-Pools gibt, müssen Sie bei allen Kommandos mit der Option --pool *poolname* angeben, auf welchen Pool Sie sich beziehen.

► `vncdisplay name`

liefert die IP-Adresse (leer für localhost) und Portnummer für die VNC-Anzeige der virtuellen Maschine. Sie können nun einen beliebigen VNC-Client starten, um mit der virtuellen Maschine zu interagieren. Aus Sicherheitsgründen funktioniert der VNC-Zugang standardmäßig nur von localhost (siehe die Datei `/etc/libvirt/qemu.conf`).

`vncdisplay` liefert kein Ergebnis, wenn die virtuelle Maschine ihr Grafiksystem gar nicht über VNC freigibt, sondern stattdessen das modernere Spice-System verwendet. In diesem Fall können Sie die virtuelle Maschine mit dem Programm `virt-viewer` bedienen. An dieses Programm können Sie direkt den Namen der virtuellen Maschine übergeben.

Sollte sich dennoch die Notwendigkeit ergeben, die Spice-Portnummer zu ermitteln, wird es schwierig. In `virsh` fehlt ein Kommando, um ähnlich wie mit `vncdisplay` den Spice-Port einer virtuellen Maschine zu ermitteln. Abhilfe schafft das folgende Kommando, das ich in einem Forum von *ubuntuusers.de* gefunden habe. Es extrahiert die Portnummer aus der Prozessliste.

```
spice_port=$(ps aux | grep vm_name | grep -oP "(?<--spice port=).*?(?=,)" )
```

## Beispiele

Nachdem Sie mit dem Kommando `virsh` die `libvirt`-Shell gestartet haben, können Sie Kommandos zur Verwaltung aller virtuellen Maschinen auf dem lokalen Rechner ausführen:

```
root# virsh
virsh# list --all
 Id Name          Status
-----
 13 fedora        laufend
  - ubuntu        ausschalten
  - windows        ausschalten
virsh# start windows
Domain windows gestartet
virsh# vncdisplay windows
:1
virsh# exit
```

Via SSH können Sie auch eine Verbindung zum Dämon `libvirtd` auf einem anderen Rechner herstellen. Wenn auf dem KVM-Host aus Sicherheitsgründen ein `root`-Login mit Passwortangabe via SSH unmöglich ist, müssen Sie vor dem ersten Verbindungsaufbau Ihren öffentlichen SSH-Schlüssel auf dem KVM-Host einrichten:

```
virsh# connect qemu qemu+ssh://user@hostname/system
user@hostname's password: *****
```

### **virt-clone** [optionen]

`virt-clone` kopiert eine heruntergefahrte virtuelle Maschine, die durch `libvirt`-Werkzeuge verwaltet wird. Dabei wird eine neue XML-Definitionsdatei sowie eine Kopie der Image-Datei erstellt. Die sonstigen Hardware-Komponenten bleiben weitgehend unverändert.

► `--auto-clone`

gibt der neuen Maschine den Namen *bisher-clone* und der neuen Image-Datei den Namen *bisher-clone.img*. Mit dieser Option kann auf die Angabe von `--name` und `--file` verzichtet werden.

► `--connect hypervisor`

stellt eine Verbindung zum angegebenen Virtualisierungssystem her.

► `-f datei` bzw. `--file=datei`

gibt den gewünschten Namen der neuen Image-Datei an. Wenn die ursprüngliche virtuelle Maschine mehrere virtuelle Festplatten besitzt, müssen Sie diese Option mehrfach verwenden. Anstelle einer Image-Datei können Sie auch den Namen einer Device-Datei angeben, z. B. wenn Sie Logical Volumes des Hostsystems als Datenspeicher verwenden.

► `--mac nn:nn:nn:nn:nn:nn`

gibt die gewünschte neue MAC-Adresse des Netzwerkadapters an. Wenn diese Angabe entfällt, verwendet `virt-clone` automatisch eine eindeutige zufällige MAC-Adresse.

► `--name name`

gibt den Namen der neuen virtuellen Maschine an.

► `--original name`

gibt den Namen der ursprünglichen virtuellen Maschine an.

► `--preserve-data`

verhindert, dass die Image-Datei kopiert wird. Die neue virtuelle Maschine verwendet also dieselbe Image-Datei wie die bisherige virtuelle Maschine. Es ist daher nicht zulässig, die alte und die neue virtuelle Maschine gleichzeitig auszuführen! Die Option ist dann zweckmäßig, wenn Sie ein System mit einer neuen virtuellen Hardware- oder Netzwerkkonfiguration testen möchten.

### Beispiel

Das folgende Kommando kopiert die Ubuntu-Server-Installation `userver5`. Die neue virtuelle Maschine erhält den Namen `userver6`, und die neue Image-Datei wird in der Datei `/var/lib/libvirt/images/userver6.img` gespeichert. Achten Sie darauf, die neue Image-Datei in einem libvirt-Speicherpool anzulegen! Andernfalls verhindern die SELinux-Regeln unter RHEL/Fedora die Ausführung der virtuellen Maschine.

```
root# virt-clone --original userver5 --name userver6 \
                --file /var/lib/libvirt/images/userver6.img
```

### **virt-install** [optionen]

`virt-install` ist ein Python-Script, das beim Einrichten neuer virtueller Maschinen hilft.

► `--arch architektur`

gibt die gewünschte CPU-Architektur an, z.B. `i386`, `i686` oder `x86_64`. Standardmäßig verwendet `virt-install` dieselbe Architektur wie auf dem Hostsystem.

► `--cdrom datei`

gibt den Dateinamen der ISO-Datei bzw. den Device-Namen des CD/DVD-Laufwerks an, von dem die Installationsdaten gelesen werden.

► `--connect hypervisor`

stellt eine Verbindung zum angegebenen Virtualisierungssystem her. Wenn `virt-install` auf einem KVM-Host mit `root`-Rechten gestartet wird, stellt das Script automatisch eine Verbindung zu `qemu:///system` her.

► `--disk datei`

gibt den Dateinamen der Image-Datei an, in der die virtuelle Maschine gespeichert werden soll. Wenn Sie mehrere virtuelle Speichergeräte verwenden möchten, müssen Sie die Option mehrfach angeben.



► `--disk opt1=wert1,opt2=wert2,...`

ermöglicht die Angabe mehrerer Image-Dateien aus unterschiedlichen libvirt-Speicherpools. Zulässige Optionen sind unter anderem `path` zur Angabe einer Image-Datei oder eines Block-Devices, `pool` zur Angabe eines zuvor eingerichteten libvirt-Speicher-Pools, `vol` zur Angabe einer bereits existierenden Image-Datei in einem Speicher-Pool (`vol=poolname/imagename`), `size` zur Angabe der gewünschten Größe neuer Image-Dateien (in GByte), `bus` zur Auswahl des Bussystems (`ide`, `scsi` oder `virtio`) und `cache` zur Auswahl des Cachings (`writethrough`, `writeback` oder `none`).

► `--graphics typ,opt1=wert1,opt2=wert2 ...`

gibt an, wie das Grafiksystem der virtuellen Maschine realisiert werden soll. Als Typ kommen `vnc` oder `spice` infrage. Standardmäßig kommt VNC zum Einsatz. Mit den weiteren Optionen können diverse Verbindungsparameter angegeben werden z. B. `port`, `listen` und `password`.

► `--import`

bewirkt, dass `virt-install` keine Neuinstallation durchführt, sondern die neue virtuelle Maschine auf Basis einer bereits existierenden Image-Datei anlegt.

► `--name name`

gibt den Namen der virtuellen Maschine an.

► `--network typ,opt1=wert1,opt2=wert2,...`

gibt die Eckdaten eines Netzwerk-Devices der virtuellen Maschine an. Die Option kann bei Bedarf mehrfach angegeben werden. `typ` gibt an, wie die Netzwerkverbindung zum Hostrechner erfolgen soll. Zulässige Werte sind `bridge=name` für eine zuvor eingerichtete Netzwerkbrücke, `network=name` für ein mit `virsh` definiertes virtuelles Netzwerk oder einfach `user`, wenn die virtuelle Maschine via NAT mit dem Hostsystem kommunizieren soll.

Die weiteren Optionen bestimmen die Parameter des Netzwerk-Devices: `model` gibt an, welcher Adapter emuliert werden soll, z. B. `e1000`, `rtl8139` oder `virtio`. `mac` bestimmt die MAC-Adresse des Devices. Wenn dieser Parameter nicht angegeben wird, generiert `virt-install` selbst eine zufällige MAC-Adresse. MAC-Adressen für KVM müssen mit `52:54:00` beginnen!

► `--nodisk`

ermöglicht eine Installation ohne Image-Datei (z. B., wenn die Installation in einen Netzwerkspeicher erfolgen soll).

- ▶ `--noreboot`  
verhindert den automatischen Neustart nach dem Abschluss der Installation.
- ▶ `--os-type name`  
gibt die Art des Betriebssystems an, das installiert werden soll, z.B. `linux`, `unix`, `windows` oder `other`. Die Information wird zur Optimierung diverser Hardware-Parameter genutzt.
- ▶ `--os-variant name`  
gibt an, welches Betriebssystem installiert werden soll, z.B. `fedora20`, `virtio26`, `rhel6` oder `win7`. Eine vollständige Liste aller bekannten Betriebssystemnamen liefert man `virt-install`. Wenn diese Option verwendet wird, kann auf die Angabe von `--os-type` verzichtet werden.
- ▶ `--ram n`  
gibt die Größe des RAMs in MByte an.
- ▶ `--serial pty`  
stattet die virtuelle Maschine mit einer seriellen Schnittstelle aus. Beim Start der virtuellen Maschine wird die serielle Schnittstelle mit einem Pseudo-TTY-Device des Hostrechners verbunden. In `virsh` können Sie den Device-Namen mit `ttyconsole vmname` ermitteln.
- ▶ `--soundhw=ac97/es1370/sb16`  
stattet die virtuelle Maschine mit einer Sound-Karte aus.
- ▶ `--vcpus=n`  
gibt die gewünschte Anzahl von CPU-Cores an (standardmäßig einer).
- ▶ `--video=cirrus/vga/vmva`  
gibt an, welcher Grafikadapter emuliert werden soll (standardmäßig `cirrus`).

### Beispiel

Bevor Sie `virt-install` ausführen, müssen Sie eine Image-Datei für die virtuelle Festplatte erzeugen:

```
root# virsh
virsh# vol-create-as default disk.qcow2 10G --format qcow2
virsh# exit
```

Mit dem Kommando `virt-install` richten Sie nun eine neue virtuelle Maschine ein:

```
root# virt-install --name myvmname --ram 1024 --cdrom install.iso \
        --os-variant rhel6 --disk vol=default/disk.qcow2 --graphics vnc \
        --noreboot
```

### **virt-top** [optionen]

`virt-top` zeigt an, wie viel CPU-Kapazität und Speicherplatz die laufenden virtuellen Maschinen benötigen. Während `virt-top` läuft, können Sie die Anzeige mit `[0]`, `[1]`, `[2]` und `[3]` nach virtuellen Maschinen, CPU-Cores, Netzwerkschnittstellen oder Datenträger-Devices ordnen. `[Q]` beendet das Kommando.

#### ► `--connect hypervisor`

stellt die Verbindung zum angegebenen Virtualisierungssystem her. Standardmäßig kommuniziert `virt-top` mit `qemu:///system`, sofern es mit `root`-Rechten gestartet wurde.

#### ► `-o sort`

gibt die gewünschte Sortierordnung an, z. B. `cpu`, `mem` oder `time`, um die Liste nach der CPU-Leistung, dem Speicherbedarf oder der gesamten Rechenzeit zu ordnen.

### Beispiel

```
root# virt-top
virt-top 14:36:17 - x86_64 8/8CPU 1600MHz 15961MB
5 domains, 3 active, 3 running, 0 sleeping, 0 paused, 2 inactive D:0 O:0 X:0
CPU: 0.1% Mem: 6144 MB (6144 MB by guests)
```

ID	S	RDRQ	WRRQ	RXBY	TXBY	%CPU	%MEM	TIME	NAME
3	R	0	22	0	0	1.3	12.0	35:20:50	kofler.info
4	R	0	0	0	0	0.1	12.0	274:52.67	pi-buch.info
1	R	0	0	0	0	0.0	12.0	39:11:52	ubuntu-buch.info
-									(centos-buch.info)
-									(michael-kofler.com)

### **virt-viewer** [optionen] name/id/uuid

`virt-viewer` ist ein VNC- und Spice-Client, der das Grafiksystem einer virtuellen Maschine anzeigt. An das Kommando wird normalerweise einfach der Name, die ID- oder die UUID-Nummer der virtuellen Maschine übergeben.

### ► `-c hypervisor`

stellt eine Verbindung zum angegebenen Virtualisierungssystem her. Die Option ist nur erforderlich, wenn es sich um einen externen KVM-Host handelt.

**volname** devicename

volname liefert den Datenträgernamen (Volume Name) einer Daten-CD oder -DVD. Beim Erzeugen eines ISO-Images mit genisoimage stellen Sie diesen Namen mit der Option -V ein.

**wait** [prozessnummer]

Das bash-Kommando wait wartet auf das Ende des angegebenen Hintergrundprozesses. Wenn keine Prozessnummer angegeben wird, wartet das Kommando auf das Ende aller laufenden Hintergrundprozesse, die von der Shell gestartet wurden.

**watch** kommando

watch führt das angegebene Kommando periodisch aus und zeigt dessen Ausgaben an.

### ► `-d` bzw. `--differences`

markiert die Unterschiede im Vergleich zum vorherigen Ergebnis des Kommandos.

### ► `-e` bzw. `--errexit`

beendet watch, wenn das aufgerufene Kommando einen Fehler zurückgibt.

### ► `-n n`

gibt an, nach wie vielen Sekunden das Kommando jeweils neu ausgeführt werden soll (standardmäßig alle zwei Sekunden).

## Beispiel

Das folgende Kommando zeigt alle Änderungen in der Liste der laufenden mysqld-Prozesse an:

```
user$ watch -d 'ps aux | grep mysqld'
```

**wc** dateien

wc zählt die Anzahl der Zeilen, Wörter und Zeichen in den angegebenen Dateien. Wenn durch Jokerzeichen mehrere Dateien erfasst werden, berechnet wc auch die Gesamtsumme der drei Angaben. wc ist auch gut für die Kombination mit anderen Programmen geeignet.

**Beispiel**

find liefert für jede reguläre Datei eine Zeile. wc zählt die Zeilen und liefert somit die Gesamtanzahl aller Dateien im Verzeichnisbaum.

```
user$ find / -type f -print | wc
```

**wget** [optionen] url1 url2 ...

wget lädt Dateien von HTTP-, HTTPS- und FTP-Servern herunter. wget kann unterbrochene Downloads wieder aufnehmen, rekursiv Links verfolgen und eignet sich zur Automatisierung von Downloads. Die zu übertragenden Dateien werden in Form von URLs (Uniform Resource Locator) angegeben, also beispielsweise als `http://meinserver.de/datei.iso`. Der Befehl wget wird durch unzählige Optionen gesteuert, von denen hier nur die wichtigsten zusammengefasst werden:

- ▶ **-b**  
führt den Download als Hintergrundprozess durch und schreibt alle Statusmeldungen in die Datei `wget-log`.
- ▶ **-B url** bzw. **--base=url**  
stellt allen relativen Links innerhalb der Datei, die Sie durch **-i datei** angegeben haben, die Basisadresse `url` voran.
- ▶ **-c**  
nimmt einen unterbrochenen Download wieder auf. Vorsicht: Wenn sich die Datei in der Zwischenzeit auf dem Server geändert hat, ist die heruntergeladene Datei fehlerhaft und enthält teilweise Daten aus der alten Datei und teilweise Daten aus der neuen Datei!
- ▶ **--force-html**  
interpretiert die mit **-i** angegebene Datei als HTML-Datei und lädt alle Dateien, auf die Links der HTML-Datei zeigen.

- ▶ `--ftp-user=user --ftp-password=pw`  
gibt den Benutzernamen und das Passwort für FTP-Downloads an. Diese Informationen können auch in der URL übergeben werden (`ftp://user:password@server/datei.txt`).
- ▶ `-i datei` bzw. `--input-file=datei`  
liest die zu übertragenden Dateien (URLs) aus der angegebenen Textdatei. Wenn Sie statt einer Datei das Zeichen `-` übergeben, erwartet `wget` die URLs aus der Standardeingabe. Die Option `-i` akzeptiert als Parameter nur lokale Dateien, keine URLs. (`-i http://server/name.html` funktioniert also nicht!)
- ▶ `--limit-rate=n`  
limitiert die Download-Menge pro Sekunde. Die Buchstaben `k` und `m` bezeichnen Kilo- bzw. Megabytes (also etwa `--limit-rate=0.25m`).
- ▶ `-q`  
verzichtet auf die Ausgabe von Statusmeldungen (*quiet*).
- ▶ `--retry-connrefused`  
unternimmt auch nach dem Fehler *connection refused* weitere Versuche, die Datei herunterzuladen. Die Option ist nur bei unzuverlässigen Download-Servern zweckmäßig, die gelegentlich aus dem Netz verschwinden und wenig später wieder auftauchen.
- ▶ `--spider`  
testet, ob alle Links in der durch `-i htmldatei --force-html` angegebenen Datei noch gültig sind. Die durch Links angegebenen Dateien werden aber nicht heruntergeladen.
- ▶ `-t n` bzw. `--tries=n`  
unternimmt bei einem Verbindungsabbruch `n` Versuche, sich neuerlich zu verbinden (standardmäßig 20). Mit `-t 0` versucht `wget` sein Glück so lange, bis der Download gelungen ist oder Sie das Kommando abbrechen.
- ▶ `-w n` bzw. `--wait=n`  
gibt an, wie viele Sekunden `wget` warten soll, bevor es die nächste Datei herunterlädt. Die Option verhindert, dass der Download-Server durch unzählige, nahezu gleichzeitige Download-Anfragen zu stark belastet wird.

## Rekursive Downloads

Die folgenden Optionen steuern rekursive Downloads:

- ▶ `-E` bzw. `--html-extension`  
fügt bei allen heruntergeladenen Dateien des Typs *application/xhtml+xml* oder *text/html*, deren Namen nicht mit *.html* oder *.htm* enden, die Endung *.html* an den Dateinamen an.
- ▶ `-H` bzw. `--span-hosts`  
verfolgt auch Links auf andere Websites.
- ▶ `-k` bzw. `--convert-links`  
ändert in den heruntergeladenen HTML-Dateien die Links so, dass sie auf die lokalen Dateien verweisen. Das ermöglicht es später, die Seiten offline anzusehen.
- ▶ `-l n` bzw. `--level n`  
limitiert die Rekursionsebene (standardmäßig 5). `-l inf` deaktiviert die Limitierung.
- ▶ `-L` bzw. `--relative`  
verfolgt nur relative Links (aber keine absoluten Links auf die Start-Website).
- ▶ `-r`  
aktiviert rekursive Downloads. Bei HTTP-Downloads verfolgt `wget` alle HTTP-Links der Startseite, lädt die so angegebenen Dateien herunter, verfolgt auch deren Links etc. `wget` berücksichtigt dabei nur relative Links sowie Links auf Seiten der Start-Website. Bei FTP-Downloads liest `wget` alle Unterverzeichnisse.
- ▶ `-p`  
lädt alle Dateien herunter, die zum Betrachten der Ausgangsdatei erforderlich sind (auch Bilder, CSS-Dateien und Dateien, bei denen die maximale Rekursionsebene sonst überschritten wäre).

## Beispiele

In der Grundform lädt `wget` die angegebene Datei einfach herunter:

```
user$ wget ftp://myftpserver.de/name.abc
```

Wenn der Download aus irgendeinem Grund unterbrochen wird, kann er mit `-c` ohne Umstände wieder aufgenommen werden:

```
user$ wget -c ftp://myftpserver.de/name.abc
```

Um eine Website später offline zu lesen bzw. um ihren aktuellen Zustand zu archivieren, hilft das folgende rekursive Download-Kommando (Option `-r`). Die Rekursionstiefe wird durch `-l 4` auf vier Ebenen limitiert.

```
user$ wget -r -l 4 -p -E -k http://website.de
```

### **whatis** datei

`whatis` gibt eine kurze Beschreibung (meist einzeilig) des angegebenen Kommandos bzw. Schlüsselworts aus. `whatis`-Beschreibungen existieren nur zu Themen, zu denen `man`-Texte installiert sind. Wenn `whatis` nicht funktioniert, fehlen wahrscheinlich die zugrunde liegenden Datenbanken, die mit `mandb` bzw. mit `makewhatis` erzeugt werden können.

### **whereis** datei

`whereis` durchsucht alle üblichen Pfade für Binärdateien, `man`-Dateien und Quellcode nach dem angegebenen Dateinamen. `whereis` ist damit weniger gründlich als `find`, dafür aber deutlich schneller. Die `man`-Seite zu `whereis` zählt auf, welche Verzeichnisse durchsucht werden.

### **which** kommando

`which` durchsucht alle in `PATH` angegebenen Pfade nach dem Kommando. `which` liefert als Antwort den vollständigen Namen des Kommandos, das ausgeführt würde, wenn das Kommando ohne Pfadinformationen aufgerufen würde. Das ist vor allem dann eine Hilfe, wenn zu einem Kommando mehrere Versionen in unterschiedlichen Verzeichnissen existieren.

In der `bash` kann statt `which` auch `type` verwendet werden. `type` hilft Ihnen dabei, herauszufinden, ob `kommando` in Wirklichkeit ein eingebautes Shell-Kommando, ein Alias oder eine selbst definierte Funktion ist.

## Beispiel

Das folgende Kommando ermittelt, wo das Kommando `ls` im Dateisystem gespeichert ist:

```
user$ which ls
/bin/ls
```



```
while bedingung; do
    kommandos
done
```

while bildet Schleifen in bash-Scripts. Die Schleife wird so lange ausgeführt, bis die angegebene Bedingung zum ersten Mal nicht mehr erfüllt ist. Das Schleifenkriterium ist der Rückgabewert des Kommandos, das als Bedingung angegeben wird. Vergleiche und Tests werden mit dem Kommando `test` oder dessen Kurzform in eckigen Klammern durchgeführt.

### Beispiele

Die folgende Schleife gibt die Zahlen 0 bis 5 aus:

```
#!/bin/bash
i=0
while [ $i -le 5 ]; do
    echo $i
    i=$((i+1))
done
```

Um eine Textdatei zeilenweise zu verarbeiten, formulieren Sie Ihr bash-Script so:

```
#!/bin/bash
while read zeile; do
    echo $zeile
    ...
done < textdatei.txt
```

```
who [optionen]
```

who zeigt eine Liste aller zurzeit eingeloggten Systembenutzer an. Auch wenn Sie allein mit Ihrem Rechner arbeiten, können Sie sich mit unterschiedlichen Namen an verschiedenen Textkonsolen einloggen. Wenn Sie wissen möchten, wer zuletzt auf diesem Rechner eingeloggt war, führen Sie das Kommando `last` aus.

#### ► -a

liefert detaillierte Informationen zu jedem Benutzer, inklusive der verwendeten Konsole und dem Login-Zeitpunkt. Bei SSH-Verbindungen zeigt `who` auch an, von welchem Rechner der Login erfolgte.

#### ► -m

gibt den Benutzernamen der gerade aktiven Konsole an. Das Kommando `who am i` hat dieselbe Bedeutung.

## Beispiel

Auf dem Testrechner gibt es neben den TTY-Prozessen für die sechs Konsolen zwei aktive Shell-Sessions des Benutzers kofler:

```
user$ who -a
          system boot  2015-09-09 08:57
          run-level 2   2015-09-09 08:57
LOGIN    tty4          2015-09-09 08:57      965 id=4
LOGIN    tty5          2015-09-09 08:57      971 id=5
LOGIN    tty2          2015-09-09 08:57      976 id=2
LOGIN    tty3          2015-09-09 08:57      977 id=3
LOGIN    tty6          2015-09-09 08:57      981 id=6
LOGIN    tty1          2015-09-09 08:57     1802 id=1
kofler   + pts/1       2015-09-24 11:45 02:02    351 (xxx.telekom.at)
kofler   + pts/2       2015-09-24 13:47 .        2354 (xxx.telekom.at)
```

```
wodim [optionen] dev=xxx isoimage
wodim [optionen] dev=xxx audiofiles
```

wodim (ehemals cdrecord) brennt Daten auf eine CD oder DVD. Bei Daten-CDs und bei DVDs müssen Sie die Daten vorher mit genisoimage als ISO-Datei aufbereiten.

Das Verhalten von wodim wird durch zahlreiche Optionen gesteuert, von denen hier die wichtigsten vorgestellt werden. Beachten Sie, dass es Optionen mit und solche ohne vorangestellten Bindestrich gibt!

- ▶ `blank=all|fast`  
löscht ein CD-RW-Medium.
- ▶ `dev=/dev/xxx`  
gibt den Device-Namen des CD- oder DVD-Laufwerks an, z.B. `/dev/hdc` oder `/dev/scd0`.
- ▶ `driver=name`  
gibt an, welcher Treiber zum Zugriff auf das CD-Laufwerk verwendet werden soll. Das ist nur selten notwendig, weil wodim den korrekten Treiber meist selbst erkennt. `wodim driver=help` liefert eine Liste der verfügbaren Treiber. Mit `wodim -checkdrive dev=n,n,n` können Sie versuchen, den Typ des vorhandenen CD-R-Laufwerks zu erkennen.
- ▶ `driveropts=burnfree`  
aktiviert einen Schutzmechanismus zur Vermeidung von Buffer-Underrun-Fehlern.

► `fs=n`

gibt die Größe des FIFO-Zwischenspeichers an, der zur Pufferung der Datenübertragung zwischen Festplatte und CD dient (z. B. `fs=8m`). Der Standardwert beträgt 4 MByte.

► `speed=n`

gibt die gewünschte Schreibgeschwindigkeit an. *n* ist ein Vielfaches der Standardgeschwindigkeit (150 kB/s bei CDs, 1385 kB/s bei DVDs). Wenn Sie auf die Angabe verzichten, berücksichtigt `wodim` die Einstellungen in `/etc/wodim.conf` oder nutzt die Standardgeschwindigkeit des Laufwerks. Mit `speed=0` verwendet `wodim` die niedrigstmögliche Geschwindigkeit, die das Laufwerk und die Medien unterstützen.

► `-audio`

aktiviert den Audio-CD-Modus. Es müssen nun Audio-Dateien als Parameter übergeben werden (üblicherweise `*.au`- oder `*.wav`-Dateien), nicht wie sonst üblich eine ISO-Datei.

► `-dao`

aktiviert den Disk-at-once-Modus. Das verhindert lästige Pausen zwischen den Tracks von Audio-CDs. Außerdem sind auf diese Weise hergestellte Daten-CDs als Master für die Massenproduktion geeignet.

Die Option funktioniert allerdings nur mit manchen Laufwerken. Zum Schreiben von DAO-CDs können Sie statt `wodim` auch das Kommando `cdrecord` verwenden.

► `-dummy`

simuliert den Schreibprozess, ohne die CD oder DVD tatsächlich zu verändern.

► `-eject`

wirft die CD nach dem Schreibprozess aus. Manche CD-Laufwerke verlangen das Auswerfen, bevor die nächste CD geschrieben werden kann, selbst wenn das Schreiben zuletzt nur mit `-dummy` simuliert wurde.

► `-format`

formatiert ein DVD+RW-Medium. Die Option wird normalerweise nicht benötigt, weil `wodim` unformatierte DVD+RW-Medien vor dem Beschreiben automatisch formatiert.

► `-multi`

gibt an, dass es sich um eine Multi-Session-CD handelt, der noch weitere Sessions hinzugefügt werden sollen. Diese Option muss bei allen Sessions mit Ausnahme der letzten Session angegeben werden.

- ▶ `-msinfo`  
liest vorhandene Session-Informationen von der CD. Die Option muss ab der zweiten Session angegeben werden.
- ▶ `-pad`  
bewirkt in Kombination mit `-audio`, dass die Länge von Audio-Dateien auf ein Vielfaches von 2352 Byte ergänzt wird (wenn das nicht ohnedies der Fall ist).
- ▶ `-scanbus`  
Mit dieser Option sucht `wodim` nach CD-R-Laufwerken am SCSI-Bus.
- ▶ `-swab`  
vertauscht die Byte-Reihenfolge der Daten. Verschiedene CD-Recorder erwarten eine unterschiedliche Reihenfolge der Bytes in Datenworten. `wodim` erkennt das selbstständig, sodass die Option im Regelfall nicht erforderlich ist.
- ▶ `-v`  
gibt ausführliche Meldungen darüber aus, was gerade geschieht.

### Beispiele

Mit den beiden folgenden Kommandos wird zuerst das Brennen einer Daten-CD simuliert (`-dummy`) und dann tatsächlich durchgeführt:

```
root# wodim -dummy -v speed=32 dev=/dev/scd0 iso.img
root# wodim -v speed=32 dev=/dev/scd0 iso.img
```

Noch schneller geht es, wenn Sie `genisoimage` und `wodim` mit einer Pipe verbinden. Dadurch sparen Sie außerdem den Platz für das ISO-Image.

```
root# genisoimage -r /master | wodim -v speed=32 dev=/dev/scd0 -
```

**write** username

`write` ermöglicht es, einem anderen Benutzer eine Nachricht zu senden. Nach der Ausführung des Kommandos werden alle eingegebenen Zeichen bis `[Strg]+[D]` zum Terminal des angegebenen Benutzers übertragen.

**xargs** kommando

`xargs` leitet die von der Standardeingabe kommenden Daten an das als Parameter angegebene Kommando weiter. `xargs` wird meist in Kombination mit einem

per Pipe vorangestellten Kommando verwendet, also in der Form `kommando1 | xargs kommando2`. Auf diese Weise können die Resultate des ersten Kommandos mit dem zweiten Kommando verarbeitet werden. Falls die Ergebnisse des ersten Kommandos so umfangreich sind, dass sie nicht in einer Kommandozeile übergeben werden können, wird `kommando2` in mehreren Schritten aufgerufen.

Der Unterschied zu `kommando1 | kommando2` besteht darin, dass `xargs` die Standardeingabe in Form von Parametern an `kommando2` übergibt. `xargs` muss eingesetzt werden, wenn `kommando2` nur die übergebenen Parameter verarbeitet. Gewöhnliche Pipes eignen sich dagegen, wenn `kommando2` die Daten aus der Standardeingabe verarbeitet.

#### ► `--null`

erwartet 0-Bytes zur Trennung von Dateinamen (nicht aber Leer- und Tabulatorzeichen). Diese Option eignet sich zur Verarbeitung von `find`-Ergebnissen, wenn `find` mit der Option `-print0` ausgeführt wurde.

### Beispiel

Das folgende Kommando durchsucht das aktuelle Verzeichnis und alle Unterverzeichnisse nach Backup-Dateien, die mit dem Zeichen `~` enden, und löscht diese. Das funktioniert auch für Dateinamen, die Leerzeichen enthalten.

```
user$ find -name '*.~' -print0 | xargs --null rm
```

### **xdpyinfo** [optionen]

`xdpyinfo` liefert im Grafikmodus umfassende Informationen zum laufenden X-Server. Mit `grep` können Sie daraus die für Sie relevanten Details extrahieren.

### Beispiel

Die folgenden Kommandos ermitteln die Release-Nummer des X-Servers, die Auflösung des Bildschirms sowie dessen Pixeldichte (Dots per Inch):

```
user$ xdpyinfo | grep release
vendor release number:    11702000
root# xdpyinfo | grep -C 1 dimensions
screen #0:
  dimensions:    1600x1200 pixels (411x311 millimeters)
  resolution:    99x98 dots per inch
```

```
xfs_admin [optionen] device/imagdatei
```

xfs\_admin zeigt bzw. verändert Parameter eines XFS-Dateisystems.

- ▶ -c 0|1  
deaktiviert bzw. aktiviert die sogenannte *Lazy-Counter*-Funktion des Dateisystems. Damit werden die Superblocks des Dateisystems seltener aktualisiert, was gewisse Dateioperationen spürbar beschleunigt.
- ▶ -f  
gibt an, dass sich das Dateisystem in einer Image-Datei und nicht auf einem Datenträger-Device befindet.
- ▶ -l  
gibt den Label des Dateisystems aus.
- ▶ -L name  
stellt den Namen (Label) des Dateisystems neu ein.
- ▶ -u  
zeigt die UUID des Dateisystems an.
- ▶ -U uuid  
weist dem Dateisystem eine neue UUID zu.

```
xfs_growfs [optionen] mount-verzeichnis
```

xfs\_growfs vergrößert ein XFS-Dateisystem. Das setzt voraus, dass das zugrunde liegende Device (z. B. eine Partition oder ein Logical Volume) vorher vergrößert wurde.

Die Vergrößerung erfolgt im laufenden Betrieb. Das Dateisystem muss also in den Verzeichnisbaum eingebunden sein. Eine Verkleinerung von XFS-Dateisystemen ist nicht vorgesehen.

xfs\_growfs kennt diverse Optionen, mit denen Sie steuern können, welche Bereiche des Dateisystems wie stark vergrößert werden sollen. Im Regelfall ist es aber nicht notwendig, diese Optionen anzugeben. xfs\_growfs erkennt, wie groß der zugrunde liegende Datenträger ist, und entscheidet selbst, wie groß die verschiedenen Bereiche des XFS-Dateisystems werden sollen.

**xfs\_info** device

`xfs_info` gibt eine Zusammenfassung der Eckdaten eines XFS-Dateisystems aus. Das Kommando ist gleichwertig zu `xfs_growfs -n`.

**xfs\_repair** [optionen] device/imagdatei

`xfs_repair` versucht ein beschädigtes XFS-Dateisystem wieder in einen konsistenten Zustand zu bringen.

## ► -d

repariert ein aktives Read-only-Dateisystem.

Normalerweise kann `xfs_repair` nur Dateisysteme reparieren, die momentan nicht genutzt werden. Diese Option ist dann zweckmäßig, wenn Sie Ihren Linux-Rechner im Single-User- oder Emergency-Modus gestartet haben und die Systempartition reparieren müssen. Nach der Reparatur müssen Sie das System neu starten. Die `man`-Seite beschreibt die Option als *dangerous*. Sicherer ist es, den Rechner mit einem Live- oder Emergency-System zu starten und die Reparatur von dort auszuführen.

## ► -f

gibt an, dass sich das Dateisystem in einer Image-Datei und nicht auf einem Datenträger-Device befindet.

## ► -n

führt keine Änderungen durch, sondern zeigt nur an, welche Reparaturen durchgeführt werden, wenn das Kommando nochmals ohne diese Option ausgeführt wird.

**xhost** +/-hostname

Mit `xhost +hostname` akzeptiert der Grafik-Server X Verbindungen vom angegebenen Hostnamen. Analog blockiert X nach `xhost -hostname` Verbindungen von diesem Host.

Aus Sicherheitsgründen erlaubt das Grafiksysteem X zumeist nur lokale Verbindungen. Damit das Arbeiten von einem externen Rechner gelingt, muss für diesen Host mit `xhost +name` explizit eine Ausnahme definiert werden. Noch liberaler ist `xhost +`: Damit wird die Zugriffskontrolle komplett deaktiviert, ein Login ist von jedem Host aus möglich.

## Beispiel

In der Praxis ist das Ausführen von `xhost` vor allem dann oft erforderlich, wenn Sie zuerst als normaler Benutzer via VNC eine Verbindung zum Grafik-Server herstellen und dann mit `sudo` oder `su` ein grafisches Programm mit `root`-Rechten ausführen möchten. Standardmäßig ist das nicht erlaubt. Abhilfe schafft `xhost +localhost` (weil ja dank VNC die Verbindung schon hergestellt wurde und es nun nur noch darum geht, Programme auch unter einem anderen Account auszuführen).

Das folgende Beispiel setzt voraus, dass Sie zuerst eine VNC-Verbindung herstellen, dann in einem Terminalfenster zuerst `xhost` ausführen und anschließend mit `sudo` ein Programm im Grafikmodus – hier den Editor `gedit` – starten.

```
user$ xhost +localhost
user$ sudo gedit          (Editor mit root-Rechten starten)
```

### `xkill` [optionen]

`xkill` beendet ein hängen gebliebenes oder halb abgestürztes X-Programm. Das betreffende Fenster muss nach dem Start des Kommandos mit der Maus angeklickt werden.

### `xrandr` [optionen]

`xrandr` ändert im laufenden Betrieb die Bildschirmauflösung, die Bildfrequenz und andere Einstellungen.

#### ► `--addmode ausgang name`

fügt für den angegebenen Display-Ausgang einen neuen, eventuell zuvor mit `--newmode` definierten Grafikmodus hinzu.

#### ► `--dpi n`

gibt an, wie viele Pixel pro Zoll angezeigt werden.

#### ► `--left-of output name` bzw. `--right-of name` bzw. `--below name` bzw. `--above name`

aktiviert den durch `--output` angegebenen Signalausgang und gibt an, wie die Bildschirme relativ zueinander positioniert sind. Das funktioniert nur, wenn in `xorg.conf` eine ausreichend große virtuelle Auflösung eingestellt wurde, damit beide Bildschirme abgedeckt werden können.



- ▶ `--newmode name freq x1 x2 x3 x4 y1 y2 y3 y4`  
definiert einen neuen Modus (eine neue Auflösung). Die Syntax entspricht der von ModeLine-Zeilen in `xorg.conf`. Zur Ermittlung der richtigen Zahlenkombination können Sie das Kommando `gtf` zu Hilfe nehmen.
- ▶ `--off`  
schaltet den mit `--output` angegebenen Signalausgang aus.
- ▶ `--orientation x`  
ändert die Bildlage. Zulässige Einstellungen sind `normal`, `inverted`, `left` und `right`.
- ▶ `--output name`  
gibt an, dass sich die weiteren Optionen auf einen bestimmten Signalausgang beziehen. Das ist zweckmäßig, wenn mehrere Monitore angeschlossen sind. Die Namen der aktiven Signalausgänge ermitteln Sie mit `xrandr -q`.
- ▶ `--primary`  
macht den mit `-output` ausgewählten Bildschirm zum primären Bildschirm, auf dem Gnome oder KDE das Panel, das Dock etc. anzeigen.
- ▶ `-q`  
gibt an, welche Einstellmöglichkeiten zur Verfügung stehen. Das Ergebnis ist stark vom eingesetzten Grafiktreiber abhängig.
- ▶ `--rate n`  
ändert die Bildfrequenz für die aktuelle Auflösung. *n* ist die gewünschte Bildfrequenz in Hertz.
- ▶ `--size n`  
ändert die Auflösung. *n* ist eine Nummer der Ergebnisliste von `xrandr -q` oder die gewünschte Auflösung in der Form *breitexhöhe*.

## Beispiel

Das folgende Kommando stellt die Auflösung auf  $1280 \times 1024$  Punkte ein:

```
user$ xrandr --size 1280x1024
```

Das nächste Kommando aktiviert sowohl den DVI- als auch den VGA-Signalausgang. (Die Namen der Ausgänge variieren je nach Grafikkarte und Treiber.) Auf beiden Monitoren wird dasselbe Bild angezeigt. Die Option `--auto` bewirkt, dass jeder Monitor in der für ihn optimalen Auflösung und Bildfrequenz betrieben wird.

```
user$ xrandr --output DVI-I-0 --auto --output VGA-0 --auto
```

Die folgenden drei Kommandos definieren einen neuen Grafikmodus in der Auflösung von 1280×720 Pixel und aktivieren diesen für den HDMI1-Ausgang:

```
user$ xrandr --newmode 1280x720 74.18 1280 1390 1430 1650 720 725 730 750
user$ xrandr --addmode HDMI1 1280x720
user$ xrandr --size 1280x720
```

### **xset** [kommando]

xset verändert diverse Einstellungen des X-Servers.

- ▶ `+dpms` bzw. `-dpms`  
aktiviert bzw. deaktiviert die Energiesparfunktionen (Display Power Management Signaling).
- ▶ `dpms n1 [n2 [n3]]`  
gibt an, nach wie vielen Sekunden die DPMS-Modi *Standby*, *Suspend* und *Off* des Monitors aktiviert werden sollen.
- ▶ `q`  
liefert eine Liste mit den aktuellen Einstellungen.
- ▶ `s n`  
aktiviert nach *n* Sekunden den Bildschirmschoner.

### **xz** [optionen] dateien

xz aus dem Paket xz-utils komprimiert die angegebenen Dateien. xz ist ein relativ neues Komprimierkommando, das noch kleinere Dateien als bzip2 liefert. Die komprimierten Dateien erhalten standardmäßig die Endung `.xz`.

- ▶ `-0` bis `-9`  
gibt an, wie gut xz komprimieren soll. Die besten Ergebnisse (also die kleinsten Dateien) liefert `-9`, allerdings ist dann xz's Bedarf an Arbeitsspeicher am höchsten. Standardmäßig gilt `-6`. In diesem Fall beansprucht xz beim Komprimieren maximal 100 MByte, beim Dekomprimieren maximal 10 MByte.
- ▶ `-d` bzw. `--decompress`  
dekomprimiert die angegebene Datei (entspricht `unxz`).

**yum** [optionen] kommando

yum installiert, aktualisiert und entfernt unter CentOS und RHEL RPM-Pakete. yum lädt die Pakete von den Paketquellen herunter, die in `/etc/yum.repos.d/*.repo` definiert sind.

Sämtliche Paketverwaltungsfunktionen werden in Form von Kommandos durchgeführt (z. B. `yum install paketname`). Unter Fedora wurde yum durch dnf ersetzt. Dieses Kommando ist zwar intern vollkommen neu implementiert, unterstützt aber im Wesentlichen dieselben Optionen und Kommandos wie yum. Die folgende Aufzählung gilt gleichermaßen für yum und dnf:

- ▶ `check-update`  
testet, ob Updates zu bereits installierten Paketen zur Verfügung stehen.
- ▶ `clean metadata`  
entfernt die Paketmetadaten aus dem Cache. Beim nächsten Start aktualisiert yum die Daten, indem es die Metadaten aller Paketquellen neu einliest.
- ▶ `clean packages`  
entfernt heruntergeladene und bereits installierte Paketdateien aus dem Cache.
- ▶ `downgrade package`  
ersetzt ein Paket durch das aus einer älteren Distribution. Die Versionsnummer der Distribution geben Sie mit `--releasever=n` an.
- ▶ `grouplist`, `groupinfo`, `groupinstall`, `groupupdate`, `groupremove`  
liefert Informationen über Paketgruppen bzw. installiert, aktualisiert oder entfernt Paketgruppen. `yum grouplist -v` listet zusammen mit jeder Paketgruppe in Klammern die dazugehörige interne englischsprachige Gruppen-ID auf.
- ▶ `history`  
liefert eine nummerierte Liste der zuletzt durchgeführten yum-Aktionen. `yum history info n` fördert Details zur Transaktion `n` zutage.
- ▶ `info name`  
liefert Informationen zum angegebenen Paket.
- ▶ `install name1 name2 ...`  
sucht die Pakete `name1`, `name2` etc. auf allen Yum-Paketquellen, lädt sie herunter und installiert sie. Gegebenenfalls werden auch weitere Pakete geladen und installiert oder aktualisiert, um Paketabhängigkeiten zu erfüllen.

- ▶ `list`  
liefert eine zweiteilige Liste. Der erste Teil enthält (in alphabetischer) Reihenfolge alle bereits installierten Pakete, der zweite Teil alle noch nicht installierten Pakete, die in den Yum-Paketquellen zur Verfügung stehen. Durch einen optionalen Parameter kann das Ergebnis auf Pakete eingeschränkt werden, deren Name einem Muster entspricht (z. B. `yum list xorg*`).
- ▶ `list available/updates/installed/extras/recent`  
schränkt die Ausgabe von `yum list` auf bestimmte Pakete ein. Beispielsweise liefert `yum list updates` eine Liste aller Pakete, zu denen Updates verfügbar sind. `yum list recent` liefert Pakete, die kürzlich in die Paketquellen aufgenommen wurden.
- ▶ `localinstall rpmfile1 rpmfile2 ...`  
installiert die angegebenen lokalen RPM-Dateien. Die Yum-Paketquellen werden nur zur Auflösung von Paketabhängigkeiten genutzt.
- ▶ `remove name1 name 2 ...`  
deinstalliert die angegebenen Pakete.
- ▶ `search suchbegriff`  
liefert eine Liste aller Pakete, die den Suchbegriff in der Beschreibung enthalten.
- ▶ `update`  
aktualisiert alle installierten Pakete.
- ▶ `update name1 name2 ...`  
aktualisiert nur die angegebenen Pakete.
- ▶ `upgrade`  
hat dieselbe Wirkung wie `update` in Kombination mit der Option `--obsoletes` (siehe unten).

Sie können das Verhalten von `yum` durch einige (selten benötigte) Optionen beeinflussen:

- ▶ `-y`  
beantwortet alle Fragen von `yum` mit *yes*. Damit kann `yum` zur Installation ohne weitere Interaktion eingesetzt werden (z. B. in einem Script).

► `--enablerepo=name`

aktiviert eine an sich deaktivierte Paketquelle. Die Option erspart es Ihnen, die `*.repo`-Datei zu ändern, wenn Sie nur ein einzelnes Paket einer nicht aktiven Paketquelle installieren möchten.

► `--exclude=paket`

schließt das angegebene Paket von der gewählten Operation (z. B. einem Update) aus.

► `--obsoletes`

bewirkt, dass yum nicht mehr benötigte Pakete nach einem Update löscht. Das ist nur bei einem Distributions-Update (Version  $n$  zu  $n+1$ ) zweckmäßig.

## Beispiele

Die beiden folgenden Kommandos aktualisieren zuerst alle bereits vorhandenen Pakete und installieren dann den Editor Emacs. Unter Fedora ersetzen Sie einfach yum durch dnf.

```
root# yum update
```

```
...
Transaction Summary
Install      1 Package(s)
Update      42 Package(s)
Remove       0 Package(s)
Total download size: 52 M
Is this ok [y/N]: y
...
```

```
root# yum install emacs
```

Package	Arch	Version	Paketquelle	Größe
Installieren:				
emacs	x86_64	1:24.3-11.el7	base	2.9 M
Als Abhängigkeiten installiert:				
ImageMagick	x86_64	6.7.8.9-10.el7	base	2.1 M
OpenEXR-libs	x86_64	1.7.1-7.el7	base	217 k
emacs-common	x86_64	1:24.3-11.el7	base	20 M
ilmbase	x86_64	1.0.3-7.el7	base	100 k
...				

Normalerweise werden Sie sich bemühen, alle installierten Pakete mit `yum/dnf update` möglichst aktuell zu halten. Gerade unter Fedora können die mitunter über-aktuellen Pakete auch Probleme bereiten. Dann kann `yum/dnf downgrade` helfen: Das folgende

Kommando ersetzt unter Fedora 23 die Pakete des Grafiksystems `xorg` durch die aus Fedora 22.

```
root# dnf --showduplicates --alloweraseing  
--releasever=22 downgrade xorg-x11-server-Xorg
```

```
yumdownloader [optionen] paketname
```

`yum` selbst ist nicht in der Lage, Quellcodepakete zu installieren. Diese Aufgabe übernimmt bei Bedarf das Kommando `yumdownloader`, das sich im Paket `yum-utils` befindet. Wenn das Kommando ohne Optionen ausgeführt wird, lädt es das angegebene Binärpaket herunter, installiert es aber nicht.

- ▶ `--resolve`  
lädt auch abhängige Pakete herunter.
- ▶ `--source`  
lädt den Quellcode (statt des Binärpakets) herunter.
- ▶ `--urls`  
zeigt die Download-Links an, lädt aber nichts herunter.

### Beispiel

Das folgende Kommando installiert den Quellcode des Gnome-Texteditors:

```
user$ yumdownloader --source gedit
```

Unter Fedora verwenden Sie anstelle von `yumdownloader` das `dnf`-Subkommando `dnf download`.

```
zcat datei.gz  
zless datei.gz  
zmore datei.gz
```

Die drei Kommandos funktionieren wie `cat`, `less` und `more`. Der einzige Unterschied besteht darin, dass mit `gzip` komprimierte Dateien direkt gelesen werden können, also ohne vorheriges Dekomprimieren mit `gunzip`.

Statt `zless` kann bei vielen Distributionen direkt `less` eingesetzt werden: Dieses Kommando ist meist so konfiguriert, dass es ohne Unterschied für komprimierte und nichtkomprimierte Dateien verwendet werden kann.

**zenity** [optionen]

zenity zeigt einfache GTK-Dialoge zur Texteingabe, zur Auswahl eines Tags im Kalender oder zur Dateiauswahl an. Das Kommando gibt das Ergebnis zur Weiterverarbeitung zurück. Mit zenity können Sie Ihren bash-Scripts einen moderneren Anstrich geben – vorausgesetzt, das Script wird in einem grafischen Desktop-System ausgeführt.

Die folgenden Optionen illustrieren lediglich das Konzept des Kommandos. Das Kommando bietet mit weit über 50 Optionen erheblich mehr Eingabe- und Steuerungsmöglichkeiten.

- ▶ `--calendar`  
zeigt einen Kalender an und ermöglicht die Auswahl eines Tags.
- ▶ `--entry`  
ermöglicht eine Texteingabe.
- ▶ `--file-selection`  
erlaubt die Auswahl einer Datei.
- ▶ `--password`  
ermöglicht eine Passworteingabe.
- ▶ `--text text`  
gibt an, welcher Text innerhalb des Dialogs angezeigt werden soll.
- ▶ `--title text`  
stattet den Dialog mit einem Titel aus.

**Beispiel**

Die folgenden Zeilen eines Scripts fordern den Benutzer zuerst zur Passworteingabe auf und zeigen das Passwort dann an:

```
pw=$(zenity --password --title 'Ihr Passwort bitte ...')
echo "Das war Ihr Passwort: $pw"
```

**zip** [optionen] archiv.zip datei1 datei2 ...

Das Kommando bildet das ZIP-Archiv `archiv.zip` und fügt darin alle angegebenen Dateien ein. Das Archiv eignet sich besonders für den Datenaustausch mit Windows-Anwendern. Um die Dateien unter Linux wieder zu extrahieren, verwenden Sie unzip.

### ► -r

archiviert rekursiv den Inhalt von Verzeichnissen.

### Beispiel

Das folgende Kommando fügt alle als Parameter übergebenen HTML-Dateien in `meinarchiv.zip` ein:

```
user$ zip meinarchiv.zip *.html
```

```
zipinfo [optionen] datei.zip
```

Das Kommando liefert das Inhaltsverzeichnis eines ZIP-Archivs.

```
zypper [optionen] kommando
```

Das SUSE-spezifische Kommando `zypper` hilft bei der RPM-Paketverwaltung. Es installiert und aktualisiert Pakete, richtet Paketquellen ein etc. `zypper` setzt die `libzypp`-Bibliothek voraus, die ein integraler Bestandteil aller aktuellen SUSE-Distributionen ist. Die im Folgenden beschriebenen Kommandos können abgekürzt werden, sofern die Bedeutung eindeutig ist, also z. B. `zypper in name` statt `zypper install name`.

### ► addrepo [optionen] uri name

richtet eine neue Paketquelle ein und gibt ihr einen Alias-Namen. Der *Uniform Resource Identifier* (URI) beschreibt die Paketquelle, beispielsweise in der folgenden Form:

```
http://download.opensuse.org/distribution/13.1/repo/oss/
```

Standardmäßig werden Paketquellen aktiviert (Spalte *enabled* in `zypper repos`) und zur Aktualisierung vorgemerkt (Spalte *refresh*). Die Paketquelle wird aber nicht erstmalig in den Cache eingelesen. Dazu müssen Sie `zypper refresh` ausführen.

Bei Paketquellen, deren Inhalt sich nicht ändert, können Sie die regelmäßige Aktualisierung durch die Option `-n` (*no refresh*) vermeiden.

Während Sie eine Paketquelle einrichten, haben Sie die Möglichkeit, den Schlüssel zu importieren, mit dem die Pakete signiert sind. Sofern Sie von der Authentizität der Paketquelle überzeugt sind, sollten Sie den diesbezüglichen Rückfragen zustimmen.



zypper addrepo bietet keine Möglichkeit, den Namen einer Paketquelle anzugeben. zypper repos zeigt in der Namensspalte deswegen nochmals den Alias an. Wenn Sie Namen und Alias getrennt einstellen möchten, müssen Sie die Datei zur Beschreibung der Paketquelle im Verzeichnis `/etc/zypp/repos.d/` direkt ändern.

- ▶ `dup`  
führt ein Distributions-Update durch, also beispielsweise von openSUSE 13.1 auf Version 13.2. Vorher müssen Sie ein reguläres Update durchführen und dann die Paketrepositories auf die neue Version umstellen. Nach dem Update ist ein Neustart erforderlich.
- ▶ `info name`  
liefert Informationen zum angegebenen Paket/Patch/Produkt etc.
- ▶ `install [-y] name`  
installiert das angegebene Paket. Statt eines Pakets darf auch der Name eines Produkts, eines Patches oder einer Sprache angegeben werden. Um alle Pakete eines vordefinierten Schemas (Patterns) zu installieren, führen Sie `zypper install -t pattern name` aus. Mit der zusätzlichen Option `-y` verzichtet zypper auf Rückfragen.
- ▶ `list-updates [-t package]`  
zeigt alle verfügbaren Patches an, mit `-t package` sowohl Patches als auch Updates.
- ▶ `patches`  
listet alle verfügbaren Patches auf.
- ▶ `patch-check`  
ermittelt die Anzahl aller verfügbaren Patches.
- ▶ `patch-info [name]`  
liefert Informationen zu allen Patches bzw. zum angegebenen Patch.
- ▶ `refresh`  
testet, ob sich die Inhaltsverzeichnisse der Paketquellen geändert haben, und liest diese gegebenenfalls neu ein – auch bei solchen Paketquellen, die mit der Option `-n` (*no refresh*) eingerichtet wurden. Mit der Option `-f` (*force*) erreichen Sie eine Neuerfassung des Inhaltsverzeichnisses selbst dann, wenn zypper glaubt, dass sich nichts geändert hat.

- ▶ `remove [-y] name`  
entfernt das angegebene Paket. Mit der zusätzlichen Option `--clean-deps` werden auch abhängige Pakete deinstalliert, die nun nicht mehr benötigt werden. Mit `-y` verzichtet zypper auf Rückfragen.
- ▶ `removereпо name`  
entfernt die durch den URI oder Alias angegebene Paketquelle.
- ▶ `renamereпо old new`  
gibt der Paketquelle einen neuen Alias-Namen.
- ▶ `repos`  
listet alle eingerichteten Paketquellen auf. Das Kommando zeigt dabei den Namen und den Alias jeder Paketquelle an, nicht aber die Adresse (*uri*). Bei mit YaST eingerichteten Paketquellen wird die Adresse als Alias verwendet, weswegen die Alias-Spalte oft wie eine Adressspalte aussieht. Für Paketquellen, die mit zypper addrepo eingerichtet werden, gilt dies aber nicht.
- ▶ `search [optionen] ausdruck`  
sucht nach Paketen mit dem angegebenen Suchausdruck. Standardmäßig werden nur die Paketnamen durchsucht. Mit `-d` durchsucht zypper auch die Paketbeschreibung. `-i` schränkt die Suche auf bereits installierte Pakete ein, `-u` auf noch nicht installierte Pakete. Mit `-t pattern` ermitteln Sie eine Liste vordefinierter Paketgruppen.
- ▶ `update [-t package] [-y]`  
aktualisiert alle Pakete, zu denen Patches verfügbar sind. Mit der Option `-t package` berücksichtigt zypper auch Updates, also neuere Programmversionen.  
  
Dazu eine kurze Erläuterung: Updates sind komplette Pakete, die in einer neueren Version als der installierten zur Verfügung stehen. Patches sind dagegen Ergänzungs- bzw. Aktualisierungspakete (Delta-RPMs), die nur die Änderungen enthalten und somit wesentlich kleiner sind.

### Beispiel

Das erste Kommando listet die Paketquellen auf, das zweite aktualisiert die Quellen, das dritte installiert den Editor nano, und das vierte stellt fest, welche Updates zur Verfügung stehen.

```
root# zypper repos
#|Alias                               |Name                               |Aktiviert|GPG-Überpr.|Aktual.
+-+-----+-----+-----+-----+-----+
1|openSUSE-42.1-0                     |openSUSE-42.1-0                     |Nein     |---        |Nein
2|packman.inode.at-suse               |Packman Repository                  |Ja       |(r ) Ja    |Ja
3|repo-oss                           |openSUSE-Leap-42.1-Oss              |Ja       |(r ) Ja    |Ja
...
root# zypper refresh
All repositories have been refreshed.
root# zypper install nano
Reading installed packages...
The following NEW package is going to be installed: nano
Overall download size: 335.0 K. After the operation, additional 1.2 M
will be used.
Continue? [YES/no]: yes
root# zypper list-updates
...
Repository:      | Name                | Version | Category   | Status
Haupt-Update Repos | MozillaFirefox      | 4572-0  | security   | Needed
Haupt-Update Repos | NetworkManager      | 4548-0  | recommended| Needed
...
```

#&%! (bash-Sonderzeichen)

Sowohl bei der Eingabe von Kommandos als auch bei der Shell-Programmierung können Sie eine unüberschaubare Fülle von Sonderzeichen für diverse Aktionen verwenden. Tabelle 2 fasst die Bedeutung der wichtigsten Sonderzeichen zusammen.

Zeichen	Bedeutung
;	trennt mehrere Kommandos.
:	Shell-Kommando, das nichts tut
.	Shell-Programm ohne eigene Subshell starten (. datei, entspricht source datei)
#	leitet einen Kommentar ein.
#!/bin/sh	identifiziert die gewünschte Shell für das Shell-Programm.
\tblcol	führt das Kommando im Hintergrund aus (kom &).
&&	bedingte Kommandoausführung (kom1 && kom2)
&>	Umleitung von Standardausgabe und -fehler (entspricht >&)
	bildet Pipes (kom1   kom2).
	bedingte Kommandoausführung (kom1    kom2)

Tabelle 2 bash-Sonderzeichen

Zeichen	Bedeutung
*	Jokerzeichen für Dateinamen (beliebig viele Zeichen)
?	Jokerzeichen für Dateinamen (ein beliebiges Zeichen)
[abc]	Jokerzeichen für Dateinamen (ein Zeichen aus abc)
[ ausdruck ]	Kurzschreibweise für test ausdruck
[[ ausdr ]]	erweiterte test-Syntax, bash-spezifisch
(...)	Kommandos in derselben Shell ausführen ((kom1; kom2))
{...}	Kommandos gruppieren
{ , , }	Zeichenketten zusammensetzen (a{1,2,3} → a1 a2 a3)
{a..b}	Zeichenketten zusammensetzen (b{4..6} → b4 b5 b6)
~	Abkürzung für das Heimatverzeichnis
>	Ausgabeumleitung in eine Datei (kom > dat)
>>	Ausgabeumleitung; an vorhandene Datei anhängen
>&	Umleitung von Standardausgabe und -fehler (entspricht &>)
2>	Umleitung der Standardfehlerausgabe
<	Eingabeumleitung aus einer Datei (kom < dat)
<< ende	Eingabeumleitung aus der aktiven Datei bis zu ende
\$	Kennzeichnung von Variablen (echo \$var)
\$_	PID des zuletzt gestarteten Hintergrundprozesses
\$\$	PID der aktuellen Shell
\$0	Dateiname des gerade ausgeführten Shell-Scripts
\$1 bis \$9	die ersten neun dem Kommando übergebenen Parameter
\$#	die Anzahl der dem Shell-Programm übergebenen Parameter
\$* oder @\$	die Gesamtheit aller übergebenen Parameter
\$_	Rückgabewert des letzten Kommandos (0 = OK oder Fehlernummer)
\$(...)	Kommandosubstitution (echo \$(ls))
\${...}	diverse Spezialfunktionen zur Bearbeitung von Zeichenketten
`\${...}`	arithmetische Auswertung (echo `\${2+3}`)
"..."	Auswertung der meisten Sonderzeichen verhindern
'...'	Auswertung aller Sonderzeichen verhindern
`...`	Kommandosubstitution (echo `ls`)
\zeichen	hebt die Wirkung des Sonderzeichens auf.

Tabelle 2 bash-Sonderzeichen (Forts.)

# Konfigurationsdateien

Dieser Abschnitt der *Linux-Kommandoreferenz* fasst die Syntax der wichtigsten Linux-Konfigurationsdateien zusammen. Die Dateien sind nach ihrem Namen geordnet, nicht nach dem Pfad des Verzeichnisses, in dem sie sich befinden.

Die meisten der hier präsentierten Dateien gelten für alle Distributionen. Auf distributionsspezifische Eigenheiten weise ich explizit hin. Weitere Details zu Konfigurationsdateien erhalten Sie mit `man 5 name`. Dabei gibt die Zahl 5 an, dass Sie die Dokumentation einer Konfigurationsdatei nachlesen möchten, nicht die eines womöglich gleichnamigen Kommandos.

## `/etc/adduser.conf`

Die Debian/Ubuntu-spezifische Datei `/etc/adduser.conf` enthält Defaulteinstellungen für das Einrichten neuer Benutzer und Gruppen durch die Kommandos `adduser` und `addgroup`. Die Datei `adduser.conf` enthält zeilenweise Einstellungen in der Form `parameter=wert`.

- ▶ `ADD_EXTRA_GROUPS=0|1`  
gibt an, ob neuen Benutzern automatisch die in `EXTRA_GROUPS` aufgezählten sekundären Gruppen zugeordnet werden sollen.
- ▶ `DHOME`  
gibt das Basisverzeichnis für die Heimatverzeichnisse an (üblicherweise `/home`).
- ▶ `DIRMODE`  
gibt an, mit welchen Zugriffsbits neue Heimatverzeichnisse eingerichtet werden sollen (standardmäßig 755).
- ▶ `DSHELL`  
gibt die Default-Shell an.
- ▶ `EXTRA_GROUPS="gruppe1,gruppe2,..."`  
enthält eine Liste von sekundären Gruppen, denen neue Benutzer standardmäßig (`ADD_EXTRA_GROUPS=1`) oder durch die `adduser`-Option `--add_extra_groups` zugeordnet werden.

- ▶ `FIRST_GID` und `LAST_GID`  
bestimmt den Bereich der GID-Nummern für gewöhnliche Gruppen.
- ▶ `FIRST_SYSTEM_GID` und `LAST_SYSTEM_GID`  
bestimmt den Bereich der GID-Nummern für die Gruppen von System-Accounts.
- ▶ `FIRST_SYSTEM_UID` und `LAST_SYSTEM_UID`  
bestimmt den Bereich der UID-Nummern für System-Accounts.
- ▶ `FIRST_UID` und `LAST_UID`  
bestimmt den Bereich der UID-Nummern für gewöhnliche Benutzer.
- ▶ `GROUPHOMES=yes|no`  
legt fest, ob die Heimatverzeichnisse von neuen Benutzern am Ort `/home/gruppenname/benutzername` eingerichtet werden sollen.
- ▶ `LETTERHOMES=yes|no`  
bestimmt, ob Heimatverzeichnisse in Unterverzeichnisse mit dem Anfangsbuchstaben platziert werden sollen, also z. B. `/home/k/kofler`. Das kann bei Rechnern mit sehr vielen Accounts zweckmäßig sein.
- ▶ `NAME_REGEX=yes|no`  
enthält einen regulären Ausdruck (siehe [`grep`](#)), dem neue Benutzernamen entsprechen müssen.
- ▶ `SKEL`  
gibt an, aus welchem Verzeichnis Defaultdateien in ein neues Benutzerverzeichnis kopiert werden sollen (in der Regel `/etc/skel`).
- ▶ `USERGROUPS=yes|no`  
gibt an, ob für jeden Benutzer eine eigene Gruppe eingerichtet werden soll.

### Beispiel

Die folgenden Zeilen zeigen auszugsweise die unter Ubuntu gültigen Defaulteinstellungen in `adduser.conf`:

```
DSHELL=/bin/bash
DHOME=/home
GROUPHOMES=no
LETTERHOMES=no
SKEL=/etc/skel
FIRST_SYSTEM_UID=100
```

```

LAST_SYSTEM_UID=999
FIRST_SYSTEM_GID=100
LAST_SYSTEM_GID=999
FIRST_UID=1000
LAST_UID=29999
FIRST_GID=1000
LAST_GID=29999
USERGROUPS=yes
USERS_GID=100
DIR_MODE=0755
...

```

### /etc/aliases

Die Datei `/etc/aliases` listet für lokale E-Mail-Accounts Weiterleitungsadressen auf. Für die Auswertung der Datei ist der E-Mail-Server verantwortlich, d. h., die Datei ist nur dann relevant, wenn auf dem Rechner ein Mail Transfer Agent (MTA) läuft, z. B. `sendmail` oder `postfix`.

Im Regelfall ist `aliases` primär für lokale Weiterleitungen gedacht, beispielsweise um E-Mails von `webmaster@hostname` an `adminxy@hostname` weiterzuleiten. Syntaktisch ist es auch zulässig, lokale Mails an externe Hosts weiterzuleiten; in der Praxis scheitert dies aber oft an Spam-Schutzmaßnahmen der externen Hosts.

Viele Mail-Server werten `/etc/aliases` nicht direkt aus, sondern berücksichtigen stattdessen eine Datenbankdatei, die aus der `aliases`-Datei generiert wird. Deswegen werden Änderungen in der `aliases`-Datei oft erst wirksam, nachdem das Kommando `newaliases` ausgeführt wird.

`/etc/aliases` enthält zeilenweise die Umleitungsregeln:

```
name: alias1, alias2, alias3 ...
```

Dabei muss `name` ein lokaler Mail-Account sein (ohne `@hostname!`). Zur Angabe des Alias gibt es mehrere Möglichkeiten, wobei es vom Mail-Server abhängt, welche Varianten unterstützt werden.

- ▶ `name`: Die E-Mail wird an einen anderen lokalen Mail-Account weitergeleitet.
- ▶ `adresse@host`: Die E-Mail wird an eine externe E-Mail-Adresse weitergeleitet.
- ▶ `/pfad/datei`: Die E-Mail wird am Ende der angegebenen Datei hinzugefügt.
- ▶ `|kommando`: Die E-Mail wird an das angegebene Kommando übergeben.
- ▶ `:include:/pfad/datei`: Die alias-Liste wird aus der angegebenen Datei gelesen.

Wenn mehrere Aliase angegeben werden, wird die E-Mail an alle Adressen weitergegeben.

### Beispiel

Im folgenden Beispiel werden E-Mails an root, postmaster und webmaster an die lokalen Benutzer michael und bernd weitergeleitet. E-Mails an support werden sowohl im support-Mailfach gespeichert als auch an michael sowie an peter.proell@firma-abc.de weitergeleitet.

```
root:      michael
postmaster: michael
webmaster: michael, bernd
support:    support, michael, peter.proell@firma-abc.de
```

### /etc/bashrc

/etc/bashrc enthält Defaulteinstellungen für die bash. Die Datei wird als bash-Script ausgeführt. Es sind daher alle von der bash unterstützten Sprachkonstrukte erlaubt. Üblicherweise wird in bashrc die Prompt-Umgebungsvariable PS1 eingestellt, bei manchen Distributionen auch der umask-Wert.

Die Einstellungen in /etc/bashrc werden durch jene in /etc/profile ergänzt. /etc/profile gilt für alle Shells, bashrc enthält nur die Einstellungen, die spezifisch für die bash gelten sollen. Zudem werden benutzerspezifische Einstellungen in .bashrc berücksichtigt.

### /boot/config.txt

config.txt ist eine Raspberry-Pi-spezifische Datei zur Konfiguration der Hardware. Sie muss sich zusammen mit anderen Bootdateien in der ersten Partition der SD-Karte befinden. Unter Raspbian ist die Datei im Verzeichnis /boot zugänglich. Die Datei wird während des Bootprozesses noch vor dem Start des Kernels direkt durch den SoC (System-on-a-Chip) des Raspberry Pi ausgewertet.

► `arm_freq=n`

gibt die gewünschte maximale CPU-Frequenz in MHz an. Beim Raspberry Pi 2 beträgt die Frequenz standardmäßig 900 MHz. Sie kann relativ gefahrlos auf 1 GHz erhöht werden (Overclocking).

► `config_hdmi_boost=n`

steuert die HDMI-Signalstärke (0 ist der normale Pegel, 7 ist der maximale Pegel).



► `core_freq=n`

gibt die Taktfrequenz der Grafik- und Encoding-Codes in MHz an (standardmäßig 250 MHz).

► `decode_MPG2=0x12345678`

`decode_WVC1=0x12345678`

gibt Codes an, um den MPG-2- und den VC-1-Decoder freizuschalten. Die Schlüssel können unter <http://www.raspberrypi.com> erworben werden und müssen zur Seriennummer der CPU passen (siehe `/proc/cpuinfo`).

► `disable_camera_led=1`

deaktiviert die Leuchtdiode, die die Aktivität der Raspberry-Pi-Kamera anzeigt.

► `disable_overscan=1`

deaktiviert die Overscan-Funktion. Solange diese aktiv ist, umgibt das Grafiksystem des Raspberry Pi das eigentliche Bild mit einem schwarzen Rand. Das erhöht die Kompatibilität zu manchen HDMI-Monitoren, führt bei anderen Modellen aber zu einem schwarzen »Trauerrand«. Abhilfe schafft dann `disable_overscan=1`. (Wenn die Overscan-Funktion erwünscht ist, können Sie mit den Parametern `overscan_left`, `-right`, `-top` und `-bottom` die Overscan-Breite auf allen vier Seiten einstellen.)

► `display_rotate=1|2|3`

rotiert die Display-Darstellung um 90, 180 bzw. 270 Grad.

► `dtoverlay=xxx`

aktiviert den Device-Tree-Overlay für eine Hardware-Komponente und lädt die entsprechenden Treibermodule.

► `dtparam=i2c_arm=on`

aktiviert den Bus I<sup>2</sup>C.

► `dtparam=spi=on`

aktiviert den SPI-Bus.

► `gpu_mem=zsine`

gibt an, wie viel RAM dem Grafiksystem zugewiesen wird. Der Raspberry Pi 2 verfügt insgesamt über 1 GByte RAM. Das RAM wird zwischen der CPU und dem Grafikprozessor geteilt. Die Aufteilung muss beim Start endgültig festgelegt werden, wobei Sie dem Grafiksystem 16, 64, 128 oder 256 MByte zuweisen können. 16 MByte sind für den normalen Betrieb ausreichend. Für grafikintensive Anwendungen (3D-Grafik, HD-Filme abspielen etc.) benötigt das Grafiksystem hingegen

128 MByte Speicher. Auch die Nutzung der Kamera erfordert ein Minimum von 128 MByte Grafikspeicher

- ▶ `hdmi_force_hotplug=1`  
aktiviert den HDMI-Ausgang auch dann, wenn kein Monitor erkannt wird.
- ▶ `hdmi_group=1|2`  
gibt an, wie die `hdmi_mode`-Einstellung zu interpretieren ist: 1 bedeutet CEA (*Consumer Electronics Association*, Modi für TV-Geräte), 2 gilt für DMT (*Display Monitor Timings* für Computer-Monitore).
- ▶ `hdmi_mode=n`  
gibt den gewünschten Display-Modus an. Dabei wird auch `hdmi_group` berücksichtigt. Die Liste der möglichen Modi können Sie mit `vservice -m CEA` bzw. `vservice -m DMT` feststellen.
- ▶ `sdram_freq=n`  
gibt die Taktfrequenz für den Arbeitsspeicher (RAM) im MHz an. Die Defaulteinstellung beträgt 400 MHz.
- ▶ `start_x=1`  
aktiviert das Kameramodul.

Veränderungen an `config.txt` werden erst mit dem nächsten Neustart wirksam. Viele `config.txt`-Parameter können Sie im laufenden Betrieb mit `vcgencmd get_config` auslesen. Noch mehr `config.txt`-Details können Sie hier nachlesen:

[http://elinux.org/RPi\\_config.txt](http://elinux.org/RPi_config.txt)

### /etc/crontab

Die Datei `/etc/crontab` wird vom Hintergrundprozess `crond` ausgewertet. `crontab` enthält zeilenweise Informationen darüber, wann welche Kommandos für welchen Benutzer-Account ausgeführt werden sollen. Das Cron-System ermöglicht es so, regelmäßig zu vorgegebenen Zeitpunkten Aufgaben automatisch durchzuführen, z. B. an jedem ersten Montag im Januar, April, Juli und Oktober um 2:30 eine Spiegelung des Dateisystems auf einen externen Backup-Server zu starten.

Jeder `crontab`-Eintrag besteht aus sieben Spalten in einer Zeile. Die Spalten sind durch Leer- oder Tabulatorzeichen voneinander getrennt. Die letzte Zeile *muss* mit einem

Zeilenumbruchzeichen abgeschlossen werden, andernfalls wird sie ignoriert. Kommentare werden mit dem Zeichen # am Zeilenanfang eingeleitet. Es ist nicht zulässig, einen Kommentar im Anschluss an einen crontab-Eintrag anzugeben.

- ▶ **Erste Spalte (Minute):** gibt an, in welcher Minute (0–59) das Programm ausgeführt werden soll.
- ▶ **Zweite Spalte (Stunde):** gibt die gewünschte Stunde an (0–23).
- ▶ **Dritte Spalte (Tag):** gibt den Tag im Monat an (1–31).
- ▶ **Vierte Spalte (Monat):** gibt den Monat an (1–12).
- ▶ **Fünfte Spalte (Wochentag):** gibt den Wochentag an (0–7, 0 und 7 bedeuten jeweils Sonntag).
- ▶ **Sechste Spalte (Benutzer/Account):** gibt an, für welchen Benutzer das Kommando ausgeführt wird (oft root).
- ▶ **Siebte Spalte (Kommando):** enthält das auszuführende Kommando. Das Kommando darf Leerzeichen enthalten; der gesamte Text bis zum Ende der Zeile wird als Kommando interpretiert.

Wenn in den ersten fünf Spalten statt einer Zahl ein \* angegeben wird, wird dieses Feld ignoriert. 15 \* \* \* \* bedeutet beispielsweise, dass das Kommando immer 15 Minuten nach der ganzen Stunde ausgeführt werden soll, in jeder Stunde, an jedem Tag, in jedem Monat, unabhängig vom Wochentag. 29 0 \* \* 6 bedeutet, dass das Kommando an jedem Samstag um 0:29 Uhr ausgeführt wird.

Für die Zeitfelder ist auch die Schreibweise \*/n erlaubt. Das bedeutet, dass das Kommando jede n-te Minute/Stunde etc. ausgeführt wird. \*/15 \* \* \* \* würde also bedeuten, dass das Kommando viertelstündlich (n:00, n:15, n:30 und n:45) ausgeführt wird.

Bei den Zeitfeldern sind weiters durch Kommata getrennte Mehrfachangaben sowie Von-bis-Bereiche zulässig. 1,13 in der zweiten Spalte bedeutet, dass das Kommando um ein Uhr morgens sowie um 13 Uhr ausgeführt werden soll. 2,4,6,8-18,20,23 in der zweiten Spalte bewirkt, dass das Kommando um 2 Uhr, 4 Uhr, 6 Uhr, 20 Uhr, 23 Uhr sowie stündlich zwischen 6 und 18 Uhr ausgeführt wird.

Anstelle der fünf Zeitspalten dürfen auch die in [Tabelle 3](#) zusammengefassten @-Kürzel verwendet werden. Eine weitere Zusatzregel besagt, dass ein Minuszeichen am Beginn der ersten Spalte verhindert, dass Syslog die Kommandoausführung protokolliert. Das ist allerdings nur erlaubt, wenn die sechste Spalte root enthält.

Kürzel	Code	Bedeutung
@reboot	–	nach jedem Reboot ausführen
@yearly	0 0 1 1 *	einmal im Jahr ausführen
@annually	0 0 1 1 *	wie @yearly
@monthly	0 0 1 * *	einmal pro Monat ausführen
@weekly	0 0 * * 0	einmal pro Woche ausführen
@daily	0 0 * * *	einmal pro Tag ausführen
@hourly	0 * * * *	einmal pro Stunde ausführen

**Tabelle 3** crontab-Intervallkürzel ersetzen die ersten fünf Spalten.

### Beispiel

Die folgenden drei Zeilen in `/etc/crontab` bewirken, dass 15 Minuten nach jeder vollen Stunde ein Maintenance-Script aufgerufen wird, dass jeden Sonntag um 0:30 ein Backup-Werkzeug gestartet wird und dass täglich um 3:15 eine Sicherheitskopie einer MySQL-Datenbank erstellt wird:

```
15 * * * * root /usr/bin/maintenance-script
30 0 * * 0 root /usr/bin/backup-tool
15 3 * * * mysql /usr/bin/mysql-backup
```

### Weitere Cron-Dateien

Neben der systemweiten Datei `/etc/crontab` bestehen je nach Distribution verschiedene weitere Möglichkeiten, Cron-Jobs zu definieren:

- **/etc/cron.d:** Ergänzend zu `/etc/crontab` wertet der Cron-Dämon `crond` bei den meisten Distributionen auch alle Dateien im Verzeichnis `/etc/cron.d` aus. Für diese Dateien gelten dieselben Syntaxregeln wie für `/etc/crontab`.
- **Benutzerspezifische crontab-Dateien:** Je nach Distribution werden im Verzeichnis `/var/spool/cron/` bzw. `/var/spool/cron/tabs` benutzerspezifische Crontab-Dateien gespeichert. Der Name der Datei gibt an, für welchen Benutzer die Datei gilt. Dafür entfällt in diesen Dateien die sechste Spalte. Zur Veränderung benutzerspezifischer Cron-Einträge ist das Kommando `crontab` vorgesehen.
- **cron.hourly, cron.daily, cron.weekly und cron.monthly:** Bei den meisten Distributionen enthält die Defaultkonfiguration in `/etc/crontab` einige Einträge, die bewirken, dass einmal pro Stunde alle Script-Dateien in `/etc/cron.hourly/*` mit root-Rechten ausgeführt werden, einmal pro Tag die Script-Dateien in `/etc/cron.daily/*` etc. Bei diesen Dateien handelt es sich um ganz gewöhnliche Script-Dateien, für die keine Crontab-Syntaxregeln gelten. Denken Sie aber daran, das `execute`-Bit zu setzen (`chmod a+x datei`)!

Die Dateinamen eigener Scripts in `cron.daily`, `cron.weekly` und `cron.monthly` dürfen ausschließlich aus Zahlen, Buchstaben und Binde- und Unterstrichen bestehen. Sobald der Dateiname auch nur einen Punkt enthält, wird das Script ignoriert! Die Scripts aus `/etc/cron.daily`, `-.weekly` und `-.monthly` werden nicht ausgeführt, wenn das Programm Anacron installiert ist.

### `/etc/deluser.conf`

Die Debian/Ubuntu-spezifische Datei `/etc/deluser.conf` enthält Defaulteinstellungen für das Löschen von Benutzern und Gruppen durch die Kommandos `deluser` und `delgroup`. Die Datei `deluser.conf` besteht aus zeilenweisen Einträgen in der Form `parameter=wert`.

- ▶ `BACKUP=0|1`  
gibt an, ob Backups von den Dateien gemacht werden sollen, die durch `REMOVE`-Parameter gelöscht werden.
- ▶ `BACKUP_TO`  
gibt den Ort des Backup-Verzeichnisses an.
- ▶ `EXCLUDE_FSTYPES="(fs1,fs2,...)"`  
gibt an, welche Dateisysteme ignoriert werden sollen, wenn nach Dateien des Benutzers gesucht wird, um ein Backup zu erstellen.
- ▶ `ONLY_IS_EMPTY=1`  
bewirkt, dass Gruppen nur gelöscht werden, wenn sie keine Mitglieder haben.
- ▶ `REMOVE_HOME=0|1`  
gibt an, ob mit dem Account auch dessen Heimatverzeichnis gelöscht werden soll.
- ▶ `REMOVE_ALL_FILES=0|1`  
gibt an, ob Benutzerdateien außerhalb des Heimatverzeichnisses gelöscht werden sollen, z. B. Spooling-Dateien und die E-Mail-Inbox.

### Beispiel

Die folgenden Zeilen zeigen die unter Ubuntu gültigen Defaulteinstellungen in `deluser.conf`:

```
REMOVE_HOME = 0
REMOVE_ALL_FILES = 0
BACKUP = 0
```

```
BACKUP_TO = "."
ONLY_IF_EMPTY = 0
EXCLUDE_FSTYPES = "(proc|sysfs|usbfs|devpts|tmpfs|afs)"
```

### /etc/dnf/dnf.conf

/etc/dnf/dnf.conf enthält Parameter für das Paketverwaltungsprogramm dnf. In Fedora 23 enthält die Datei nur drei Einstellungen:

```
# Datei /etc/dnf/dnf.conf unter Fedora 23
[main]
gpgcheck=1
installonly_limit=3
clean_requirements_on_remove=true
```

Alle im obigen Listing enthaltenen Schlüsselwörter haben die gleiche Bedeutung wie in /etc/yum.conf (siehe dort). dnf verwendet auch dieselben Paketquellen wie yum. Die Konfiguration der Paketquellen wurde deswegen von yum übernommen und befindet sich in \*.repo-Dateien im Verzeichnis /etc/yum.repos.d. Details können Sie ebenfalls bei der Konfigurationsdatei /etc/yum.conf nachlesen.

### /etc/fstab

/etc/fstab enthält zeilenweise Einträge für alle Dateisysteme und Swap-Partitionen, die beim Rechnerstart in den Verzeichnisbaum integriert bzw. aktiviert werden sollen. Jeder Eintrag besteht aus sechs Spalten, die durch Leer- oder Tabulatorzeichen voneinander getrennt sind. Kommentare werden mit dem Zeichen # eingeleitet.

- **Die erste Spalte (Device)** enthält den Device-Namen des Datenträgers. Statt des Device-Namens können Sie mit LABEL=xxx oder mit UUID=xxx auch den Namen bzw. die ID-Nummer des Dateisystems angeben.
- **Die zweite Spalte (Pfad)** gibt an, bei welchem Verzeichnis der Datenträger in den Dateibaum eingebunden wird. Die in der zweiten Spalte angegebenen Verzeichnisse müssen bereits existieren. Bei Swap-Partitionen geben Sie hier das Schlüsselwort none an.
- **Die dritte Spalte (Dateisystemtyp)** gibt das Dateisystem an, z. B. ext4, btrfs oder vfat. Einen Überblick über die wichtigsten Linux-Dateisysteme finden Sie bei der Beschreibung des Kommandos mount. Mit dem Eintrag auto versucht Linux, das Dateisystem selbst zu erkennen. Bei Swap-Partitionen, die intern kein Dateisystem enthalten, geben Sie das Schlüsselwort swap an.
- **Die vierte Spalte (Optionen)** enthält die gewünschten mount-Optionen. Eine Referenz der Optionen für alle wichtigen Dateisysteme finden Sie in diesem Buch

beim `mount`-Kommando. Wenn Sie keine Optionen benötigen, geben Sie stattdessen das Schlüsselwort `defaults` an. Mehrere Optionen werden nur durch Kommata getrennt, nicht durch Leerzeichen!

- **Die fünfte Spalte (`dump`)** enthält Informationen für das Unix-Programm `dump`. Unter Linux wird diese Spalte ignoriert. Es ist üblich, für die Systempartition 1 und für alle anderen Partitionen oder Datenträger 0 einzutragen.
- **Die sechste Spalte (`fsck`)** gibt an, ob und in welcher Reihenfolge die Dateisysteme beim Systemstart überprüft werden sollen. Oft wird 1 für die Systempartition und 0 für alle anderen Partitionen eingetragen. Das bedeutet, dass beim Rechnerstart nur die Systempartition auf Fehler überprüft und gegebenenfalls durch `fsck` repariert wird. Falls Sie möchten, dass weitere Partitionen automatisch überprüft werden, geben Sie bei diesen Partitionen die Ziffer 2 an. Das heißt, die Überprüfung soll nach der Kontrolle der Systempartition erfolgen. Wenn Einträge in der fünften und sechsten Spalte in `/etc/fstab` fehlen, wird 0 angenommen.

### Beispiel

Das folgende Beispiel zeigt die `fstab`-Datei eines Servers mit LVM und RAID. Die Bootpartition befindet sich auf dem RAID-Device `/dev/md/0`. Die System- und Datenverzeichnisse nutzen Logical Volumes als Speicherort. Die Dateisysteme für `/`, `/var` und `/backup` werden beim Neustart überprüft. Sollte im Root-Dateisystem ein Fehler auftreten, wird das Dateisystem im Read-only-Modus eingebunden.

<code>/dev/md/0</code>	<code>/boot</code>	<code>ext2</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/vg0/root</code>	<code>/</code>	<code>ext4</code>	<code>errors=remount-ro</code>	<code>0</code>	<code>1</code>
<code>/dev/vg0/var</code>	<code>/var</code>	<code>ext4</code>	<code>defaults</code>	<code>0</code>	<code>2</code>
<code>/dev/vg0/backup</code>	<code>/backup</code>	<code>ext4</code>	<code>defaults</code>	<code>0</code>	<code>2</code>
<code>/dev/vg0/swap</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>

### `/etc/group`

`/etc/group` enthält eine Liste aller Gruppennamen mit den zugehörigen Gruppenidentifikationsnummern (GIDs) sowie mit einer Aufzählung aller Benutzer, die der Gruppe angehören. Zur Administration der Gruppen verwenden Sie üblicherweise die Kommandos `groupadd`, `groupmod` und `groupdel`.

Die zeilenweisen Einträge in der `group`-Datei bestehen aus vier Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- **Die erste Spalte (Gruppenname)** gibt den Namen der Gruppe an.
- **Die zweite Spalte (GID)** enthielt in älteren Linux-Versionen den Hash-Code des Gruppenpassworts. Aktuelle Linux-Distributionen speichern hier das Zeichen `x`.

Sofern Gruppenpasswörter definiert sind, werden diese in /etc/gshadow gespeichert.

- ▶ **Die dritte Spalte (GID)** enthält die Gruppenidentifikationsnummer. Bei den meisten Distributionen sind GIDs kleiner 1000 für System-Accounts reserviert.
- ▶ **Die vierte Spalte (Accounts)** enthält eine durch Kommata getrennte Liste aller Benutzer, die Mitglied dieser Gruppe sind. Dabei werden allerdings nur sekundäre Mitgliedschaften berücksichtigt. Die primäre Gruppe jedes Benutzers wird hingegen in der vierten Spalte von /etc/passwd gespeichert.

### Beispiel

Die folgenden Zeilen zeigen Ausschnitte aus /etc/group auf einem Rechner, der als Web- und Entwicklungs-Server dient. Jeder System-Account und jeder reguläre Benutzer hat seine eigene Gruppe. Die Mitglieder der `admin`-Gruppe dürfen mit `sudo` Administrationsarbeiten erledigen. Die Mitglieder der `devel`-Gruppe haben besondere Zugriffsrechte auf das Versionsverwaltungssystem.

```
root:x:0:
bin:x:1:
admin:x:109:kofler,huber
...
kofler:x:1000:
huber:x:1001:
mueller:x:1002:
devel:x:1023:kofler,huber,mueller,gruber,schmiedt
...
```

### /etc/default/grub

/etc/default/grub enthält Defaulteinstellungen für den Linux-Bootloader GRUB 2. Diese Einstellungen werden erst berücksichtigt, wenn die GRUB-Konfigurationsdatei /boot/grub/grub.cfg neu erzeugt wird – entweder automatisch bei einem Kernel-Update oder manuell durch das Kommando update-grub (Debian, Ubuntu) bzw. grub-mkconfig (Fedora, openSUSE).

Die folgende Auflistung beschreibt die wichtigsten Schlüsselwörter in /etc/default/grub:

- ▶ `GRUB_CMDLINE_LINUX` und `GRUB_CMDLINE_LINUX_DEFAULT` werden von `10_linux` berücksichtigt und geben an, welche Optionen an den Kernel übergeben werden sollen. Die `GRUB_CMDLINE_LINUX`-Optionen gelten für jeden Start; die `GRUB_CMDLINE_LINUX_DEFAULT`-Optionen werden zusätzlich für den Standardstart hinzugefügt, aber nicht für den Recovery Mode.



► `GRUB_DEFAULT`

gibt an, welcher GRUB-Menüeintrag standardmäßig ausgewählt werden soll. Die Einstellung "saved" bedeutet, dass der zuletzt ausgewählte Menüeintrag aktiviert wird. Das funktioniert allerdings nur, wenn sich die GRUB-Dateien in einer gewöhnlichen Partition befinden! Ist dagegen LVM oder RAID im Spiel, kann GRUB nach der Menüauswahl keine Umgebungsvariablen speichern.

Eine weitere Möglichkeit besteht darin, `GRUB_DEFAULT` die `menuentry`-Zeichenkette des gewünschten Menüeintrags zuzuweisen. Dabei müssen Sie aber darauf achten, die Schreibweise exakt einzuhalten.

► `GRUB_DISABLE_LINUX_UUID=true`

bewirkt, dass GRUB das Root-Verzeichnis als Device-Namen an den Kernel übergibt. Ohne diese Option übergibt GRUB die UUID-Nummer des Dateisystems. Diese Einstellung gilt nur für den Start der aktiven Distribution (Script `10_linux`), nicht für andere Distributionen.

► `GRUB_DISABLE_RECOVERY=true`

verhindert, dass `update-grub` bzw. `grub2-mkconfig` Menüeinträge zum Start von Linux im Recovery Mode in `grub.cfg` einbaut.


► `GRUB_DISTRIBUTOR`

wird vom Script `/etc/grub.d/10_linux` ausgewertet und gibt den Namen der aktuellen Distribution an.

► `GRUB_GFXMODE`

gibt an, in welchem Grafikmodus GRUB laufen soll (standardmäßig in einer Auflösung von  $640 \times 480$  Pixel).

► `GRUB_HIDDEN_TIMEOUT`

ist dann von Bedeutung, wenn GRUB während der Installation nur eine einzige Linux-Distribution auf Ihrem Rechner erkennt. In diesem Fall gibt `GRUB_HIDDEN_TIMEOUT` an, wie lange der Benutzer Zeit hat, um mit  das GRUB-Menü anzuzeigen. Während dieser Wartezeit bleibt der Bildschirm schwarz. Wenn mehrere Betriebssysteme installiert sind, ignoriert GRUB die `GRUB_HIDDEN_TIMEOUT`-Einstellung und zeigt das Menü an.

► `GRUB_HIDDEN_TIMEOUT_QUIET=true`

verhindert, dass während der `GRUB_HIDDEN_TIMEOUT`-Wartezeit ein Countdown-Zähler angezeigt wird. Wenn Sie möchten, dass das GRUB-Menü immer angezeigt

wird, stellen Sie den Zeilen `GRUB_HIDDEN_TIMEOUT=...` und `GRUB_HIDDEN_TIMEOUT_QUIET=...` jeweils das Kommentarzeichen `#` voran.

► `GRUB_TIMEOUT=n`

gibt an, wie viele Sekunden GRUB auf die Auswahl eines Menüeintrags wartet. Wenn diese Zeit ohne Benutzereingaben verstreicht, startet GRUB das ausgewählte Betriebssystem. Die hier eingestellte Zeit kommt nur zur Geltung, wenn das GRUB-Menü überhaupt erscheint.

► `GRUB_TERMINAL=console`

bewirkt, dass GRUB im Textmodus läuft. Dort können allerdings keine Unicode-Zeichen angezeigt werden.

### Beispiel

In Ubuntu enthält die Konfigurationsdatei die folgenden Einstellungen:

```
# Datei /etc/default/grub
GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

### `/boot/grub/grub.cfg`

`/boot/grub/grub.cfg` bzw. `/boot/grub2/grub.cfg` ist die Konfigurationsdatei der aktuellen GRUB-Version 2. Diese Datei wird automatisch generiert, direkte Veränderungen sind nicht empfehlenswert. Um Einfluss auf das Verhalten von GRUB zu nehmen, können Sie in `/etc/default/grub` diverse Defaulteinstellungen verändern oder in `/etc/grub.d/*custom` Anweisungen einfügen, die dann direkt in `grub.cfg` übernommen werden. Um `grub.cfg` anschließend neu zu erstellen, führen Sie eines der folgenden Kommandos aus:

root#	<code>update-grub</code>	(Debian und Ubuntu)
root#	<code>grub2-mkconfig -o /etc/grub2.cfg</code>	(Fedora mit BIOS)
root#	<code>grub2-mkconfig -o /etc/grub2-efi.cfg</code>	(Fedora mit EFI)
root#	<code>grub2-mkconfig -o /boot/grub2/grub.cfg</code>	(openSUSE mit BIOS)
root#	<code>grub2-mkconfig -o /boot/grub2-efi/grub.cfg</code>	(openSUSE mit EFI)

**/etc/gshadow**

Die Datei `/etc/gshadow` enthält die Hashcodes der Gruppenpasswörter sowie weitere Daten zur Gruppenadministration. Aus Sicherheitsgründen darf nur `root` diese Datei lesen und verändern.

Die zeilenweisen Einträge in der `gshadow`-Datei bestehen aus vier Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- ▶ **Die erste Spalte (Gruppenname)** gibt den Namen der Gruppe an.
- ▶ **Die zweite Spalte (Hash-Code)** enthält den Hash-Code des Gruppenpassworts oder einen Stern (kein Passwort). Die Verwendung von Gruppenpasswörtern ist aus Sicherheitsgründen unüblich. Wenn Sie sich doch dafür entscheiden, legen Sie das Gruppenpasswort mit dem Kommando `gpaswd` fest. Jeder, der das Gruppenpasswort kennt, kann nun mit `newgrp` in diese Gruppe wechseln, auch wenn er nicht Mitglied der Gruppe ist.
- ▶ **Die dritte Spalte (Administratoren)** enthält eine durch Kommata getrennte Aufzählung von Gruppenadministratoren (Account-Namen). Gruppenadministratoren dürfen das Gruppenpasswort ändern und andere Mitglieder zur Gruppe hinzufügen bzw. wieder aus ihr entfernen.
- ▶ **Die vierte Spalte (Mitglieder)** enthält eine durch Kommata getrennte Aufzählung von Gruppenmitgliedern, die die Gruppe *ohne* Passwortangabe aktivieren dürfen.

Auf vielen Linux-Systemen ist die Datei `gshadow` ungenutzt und sieht dann so aus:

```
root:::
daemon:::
bin:::
sys:::
...
```

**/etc/host.conf**

`/etc/host.conf` steuert die Resolver-Bibliothek, die bei Netzwerkopoperationen angibt, wie unbekannte Hostnamen aufgelöst werden.

- ▶ `multi on|off`  
gibt an, ob die Resolver-Bibliothek für jeden Host alle Einträge (on) oder nur den ersten passenden Eintrag zurückgibt (off).
- ▶ `order a,b,c...`  
gibt an, in welcher Reihenfolge welche Verfahren zur Hostname-Ermittlung genutzt werden sollen. `bind` bedeutet, dass die Resolver-Bibliothek auf den in

/etc/resolv.conf eingestellten Nameserver zurückgreift. `hosts` lässt die Auswertung von /etc/hosts zu. `nis` bewirkt, dass der veraltete Network Information Service zum Einsatz kommt.

### Beispiel

Die Standardkonfiguration für `host.conf` sieht bei den meisten Distributionen so wie im folgenden Listing aus. Die Resolver-Bibliothek wertet zuerst die Datei `/etc/hosts` aus und kontaktiert dann den in `resolv.conf` eingestellten Nameserver. Die `multi`-Zeile erlaubt, dass einem in `/etc/hosts` angegebenen Hostnamen mehrere IP-Adressen zugeordnet werden dürfen.

```
order hosts, bind
multi on
```

### **/etc/hostname**

Die Datei `/etc/hostname` enthält bei vielen Linux-Distributionen den vollständigen Namen des Rechners, der aus dem Host- und dem Domainnamen besteht. `/etc/hostname` auf der Ubuntu-Server-Installation für meinen Webserver enthält daher diesen Text:

```
kofler.info
```

`/etc/hostname` wird von Debian, Fedora, Ubuntu sowie von allen Distributionen mit einer aktuellen Systemd-Version ausgewertet. In RHEL bis Version 6 wird der Hostname hingegen in `/etc/sysconfig/network` eingestellt, bei SUSE-Distributionen in `/etc/HOSTNAME`.

`/etc/hostname` wird beim Rechnerstart ausgewertet. Änderungen im laufenden Betrieb müssen mit dem Kommando `hostname` durchgeführt werden. Bei aktuellen Distributionen mit dem Init-System Systemd kann der Hostname dynamisch und dauerhaft durch das Kommando `hostnamectl` eingestellt werden.

### **/etc/hosts**

`/etc/hosts` ordnet IP-Adressen ihren Hostnamen zu. Die Datei enthält normalerweise nur Einträge für `localhost` sowie bei statisch konfigurierten Servern für den Hostnamen. In der Vergangenheit wurde die Datei verwendet, um andere Rechner im lokalen Netzwerk aufzulisten. Diese Funktion übernimmt heute aber selbst in kleinen Netzen ein Router bzw. ein lokaler Nameserver, z. B. Bind oder Dnsmasq.

Die Datei `/etc/hosts` enthält zeilenweise Einträge gemäß dem folgenden Muster:

```
ip-addr  hostname1 hostname2 hostname3 ...
```

Die Minimalvariante sieht so aus:

```
127.0.0.1  localhost
```

Bei den meisten Linux-Distributionen ist `localhost` auch für IPv6 definiert. Die folgenden Zeilen zeigen die Defaulteinstellungen unter Fedora und RHEL:

```
127.0.0.1  localhost localhost.localdomain localhost4 localhost4.localdomain4
::1       localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Bei einer statischen Netzwerkkonfiguration, z. B. auf einem Root-Server, kann `hosts` auch einen Eintrag mit der IP-Adresse und dem Hostnamen des Rechners enthalten:

```
211.212.213.214  firma-abc.de firma-abc
```

### `/etc/sysconfig/network-scripts/ifcfg-xxx`

Auf Fedora- und Red-Hat-Systemen enthält die Datei `/etc/sysconfig/network-scripts/ifcfg-xxx` die Netzwerkkonfiguration für die Schnittstelle `xxx`. Die Einstellungen der Schnittstelle `eth0` befinden sich somit in `ifcfg-eth0`. Die Datei enthält zeilenweise Variablenzuweisungen. Bei einfachen Zeichenketten ohne Leerzeichen sind die Anführungszeichen optional.

Die Syntax der `ifcfg`-Datei sowie anderer Red-Hat-spezifischer Konfigurationsdateien im Verzeichnis `/etc/sysconfig` ist in der Datei `/usr/share/doc/ini-scripts-n/network-scripts-n/network-scripts-n/sysconfig.txt` dokumentiert. Es gibt aber keine entsprechenden `man`-Seiten.

### Grundeinstellungen

- ▶ `DEVICE="devname"`  
enthält den Schnittstellennamen, z. B. `eth0` oder `enp0s5`.
- ▶ `HWADDR="nn:nn:nn:nn:nn:nn:nn"`  
gibt die MAC-Adresse der Schnittstelle an.
- ▶ `NM_CONTROLLED="yes|no"`  
gibt an, ob die Schnittstelle durch den NetworkManager gesteuert werden soll. Alle weiteren Einstellungen werden dann ignoriert! Die Defaulteinstellung lautet `yes`. Wenn Sie die Schnittstelle manuell in `ifcfg-xxx` konfigurieren möchten, muss die Datei unbedingt `NM_CONTROLLED="no"` enthalten!

- ▶ `ONBOOT="yes|no"`

gibt an, ob die Schnittstelle beim Rechnerstart automatisch aktiviert werden soll.

- ▶ `TYPE="Ethernet"`

gibt die Art der Schnittstelle an. Außer Ethernet sind z.B. Bridge für Netzwerkbrücken oder IPsec für IP-Secure-Schnittstellen zulässig. An dieser Stelle berücksichtige ich allerdings nur die Konfiguration von Ethernet-Schnittstellen.

### IPv4-Konfiguration

- ▶ `BOOTPROTO="dhcp|none"`

gibt an, ob die Konfiguration der Schnittstelle durch DHCP oder statisch erfolgt. Die weiteren Parameter sind nur für `BOOTPROTO="none"` relevant.

- ▶ `IPADDR="n.n.n.n"`

bestimmt die IP-Adresse.

- ▶ `PREFIX="n"`

legt das Netzwerk-Präfix fest. `PREFIX="24"` entspricht `NETMASK="255.255.255.0"`.

- ▶ `NETMASK="n.n.n.n"`

legt die Netzwerkmaske fest (nur, wenn eine `PREFIX`-Angabe fehlt).

- ▶ `NETWORK="n.n.n.n"`

definiert die Netzwerkadresse. Der Parameter gilt als veraltet, weil die Netzwerkadresse aus `IPADDR` und `NETMASK` automatisch ermittelt werden kann.

- ▶ `BROADCAST="n.n.n.n"`

gibt die Broadcast-Adresse an. Der Parameter gilt als veraltet, weil die Broadcast-Adresse aus `IPADDR` und `NETMASK` automatisch ermittelt werden kann.

- ▶ `GATEWAY="n.n.n.n"`

gibt die Gateway-Adresse an. Diese Adresse kann alternativ auch in der Datei `/etc/sysconfig/network` eingestellt werden, wenn es ein zentrales Gateway für mehrere Schnittstellen gibt.

- ▶ `ZONE="name"`

gibt an, welcher FirewallD-Zone die Schnittstelle zugewiesen ist. Die Einstellung betrifft nur Distributionen, die FirewallD verwenden, z.B. Fedora ab Version 19. Wenn die Einstellung fehlt, gelten für die Schnittstelle die Regeln der Default-Firewall-Zone. Diese Zone können Sie mit `firewall-cmd --get-default-zone` ermitteln.

## IPv6-Konfiguration

- ▶ `IPV6INIT="yes|no"`  
aktiviert IPv6 für diese Schnittstelle (Default no).
- ▶ `IPV6_AUTOCONF="yes|no"`  
wertet das Router Advertisement des IPv6-Routers aus und führt eine automatische Konfiguration durch (Default yes).
- ▶ `DHCPV6C="yes|no"`  
bezieht die IP-Konfiguration von einem DHCPv6-Server (Default no).
- ▶ `DHCPV6C="-x -y"`  
enthält Optionen für den DHCP-Client. Eine Liste der zulässigen Optionen liefert man `dhclient`.
- ▶ `IPV6ADDR="nnn"`  
gibt die IPv6-Adresse statisch an.
- ▶ `IPV6_DEFAULTGW="nnn"`  
gibt die Gateway-Adresse für IPv6 statisch an.

## Beispiele

Auf Desktop-Systemen kümmert sich in der Regel der NetworkManager um die Netzwerkkonfiguration. Die einzige Voraussetzung besteht darin, dass `ifcfg-xxx` *nicht* die Zeile `NM_CONTROLLED="no"` enthält. Alle weiteren Parameter trägt der NetworkManager selbst in `ifcfg-xxx` ein.

Wenn Sie die IPv4-Konfiguration selbst durchführen und es einen DHCP-Server im lokalen Netzwerk gibt, sieht `ifcfg-xxx` typischerweise so aus:

```
DEVICE=enp0s5
HWADDR=00:1C:42:85:09:A1
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=dhcp
TYPE=Ethernet
ZONE="trusted"
```

Bei einer statischen Konfiguration muss die Datei dem folgenden Muster entsprechen:

```
DEVICE=eth0
HWADDR=00:1C:42:85:09:A1
```

```
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=none
TYPE=Ethernet
IPADDR=10.0.17.33
NETMASK=255.255.255.0
GATEWAY=10.0.17.1
ZONE="trusted"
```

Wenn Sie IPv6 nutzen möchten, ergänzen Sie die `ifcfg`-Datei um die folgenden Zeilen, damit die Schnittstelle das *Router Advertisement* des IPv6-Routers auswertet und die IPv6-Parameter automatisch einstellt:

```
...
IPV6INIT=yes
IPV6_AUTOCONF=yes
```

Wenn die IPv6-Parameter von einem DHCPv6-Server bezogen werden sollen, sieht die Konfiguration so aus:

```
...
IPV6INIT=yes
DHCPV6C=yes
```

Zu guter Letzt folgt hier noch die Variante für eine statische IPv6-Konfiguration:

```
...
IPV6INIT=yes
IPV6ADDR=2a01:4f8:161:107::2/64
IPV6_DEFAULTGW=fe80::1
```

### **/etc/inittab**

Die Datei `/etc/inittab` enthält Defaulteinstellungen für das klassische Init-V-System. Die Datei ist nur für Distributionen relevant, die dieses Init-System verwenden, z. B. für Debian 7 und darauf basierende Raspbian-Versionen sowie für RHEL 5. Aktuelle Linux-Distributionen verwenden als Init-System Systemd, wo es keine mit `inittab` vergleichbare zentrale Konfigurationsdatei gibt.

Für die Zeilen der `inittab`-Datei gilt die folgende Syntax:

```
id-code:runlevel:action:command
```

`id-code` besteht aus zwei Zeichen, die die Zeile eindeutig identifizieren. Der `runlevel` gibt an, für welchen Runlevel der Eintrag gilt. `action` enthält eine Anweisung für `init`. `command` gibt an, welches Linux-Kommando oder Programm gestartet werden soll. Tabelle 4 zählt die wichtigsten `action`-Schlüsselwörter auf. Eine vollständige Beschreibung erhalten Sie mit `man inittab`.



Schlüsselwort	Bedeutung
ctrlaltdel	gibt an, wie init auf <code>[Strg]+[Alt]+[Entf]</code> reagieren soll.
initdefault	definiert den Standard-Runlevel für init (siehe oben).
once	init startet das angegebene Kommando beim Runlevel-Wechsel.
respawn	init startet das Kommando nach seinem Ende wieder neu.
sysinit	init startet das Kommando einmal während des Bootprozesses.
wait	init wartet auf das Ende des nachfolgenden Kommandos.
bootwait	init startet den Prozess während des Bootprozesses und wartet auf das Ende des nachfolgenden Kommandos.

**Tabelle 4** »inittab-action«-Schlüsselwörter (dritte Spalte)

### Beispiel

Das folgende Listing gibt die leicht gekürzte inittab-Datei von Debian 7 wieder. Als Standard-Runlevel gilt 2. Bei einem normalen Systemstart führt init die Script-Dateien rcS und das Kommando rc 2 aus. Schließlich wird für die Textkonsolen 1 bis 6 das Programm `mingetty` gestartet, das einen Login ermöglicht.

```
# Datei /etc/inittab bei Debian 7
# Standard-Runlevel
id:2:initdefault:

# Systemkonfiguration und -initialisierung unmittelbar nach dem Rechnerstart
si::sysinit:/etc/init.d/rcS

# Verhalten im Single-User-Modus (Kernelparameter su)
~~:S:wait:/sbin/sulogin

# Start der jeweiligen Runlevel
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
12:2:wait:/etc/init.d/rc 2
13:3:wait:/etc/init.d/rc 3
14:4:wait:/etc/init.d/rc 4
15:5:wait:/etc/init.d/rc 5
16:6:wait:/etc/init.d/rc 6
# Die folgende Zeile sollte nie erreicht werden, sie ist nur für Notfälle da.
z6:6:respawn:/sbin/sulogin

# Reaktion auf Strg+Alt+Entf in einer Textkonsole
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now

# Reaktion auf einen Stromausfall, der von einer unterbrechungsfreien
# Stromversorgung gemeldet wird
```

```
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
```

```
# gettys (Terminalemulatoren) für die Textkonsolen starten
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

### /etc/inittab bei CentOS 6 bzw. RHEL 6

Red Hat Enterprise Linux 6 verwendet zwar Upstart als Init-System, dennoch gibt es dort noch `/etc/inittab`. Ausgewertet wird allerdings nur die mit `id:` beginnende Zeile, die den Default-Runlevel bestimmt. Das folgende Beispiel gilt für eine Desktop-Installation, wo standardmäßig der Runlevel 5 gestartet wird, also die grafische Benutzeroberfläche.

```
id:5:initdefault:
```

### /etc/network/interfaces

Die Datei `/etc/network/interfaces` enthält bei Debian- und Ubuntu-Distributionen die Einstellungen für die Netzwerkkonfiguration. Bei Desktop-Installationen ist die Datei normalerweise fast leer und enthält nur die Einstellungen für die Loopback-Schnittstelle:

```
# Loopback-Schnittstelle
auto lo
    iface lo inet loopback
```

Für alle weiteren Schnittstellen ist der NetworkManager verantwortlich.

### IPv4-Konfiguration

Ganz anders sieht es bei Server-Installationen bzw. auf Rechnern aus, deren Netzwerkkonfiguration statisch eingestellt ist. In diesem Fall leitet die Zeile `auto name` einen Konfigurationsblock für jede Netzwerkschnittstelle ein, die beim Rechnerstart aktiviert werden soll. Die folgende Zeile `iface name options` beschreibt die Basis-konfiguration der Schnittstelle. Je nach Konfigurationstyp folgen dann die weiteren Parameter. Wenn die Schnittstelle die IP-Daten von einem DHCP-Server bezieht, fällt die Konfiguration sehr knapp aus:

```
# IPv4-Netzwerkschnittstelle mit DHCP
auto eth0
iface eth0 inet dhcp
```

Bei einer statischen Konfiguration werden die Netzwerkparameter zeilenweise durch mehrere Schlüsselwörter angegeben, deren Bedeutung selbsterklärend ist. Das Schlüsselwort `dns-nameservers` dient nur unter Ubuntu zur Konfiguration der Nameserver. Für die Auswertung ist das Paket `resolvconf` zuständig. Unter Debian müssen Sie den Nameserver hingegen in `/etc/resolv.conf` selbst einstellen.

```
# IPv4-Netzwerkschnittstelle, statische Konfiguration
auto eth0
iface eth0 inet static
    address      211.212.213.37
    netmask      255.255.255.224
    gateway      211.212.213.1
    dns-nameservers 211.222.233.244 212.232.234.245
```

Änderungen an einer einzelnen Schnittstelle aktivieren Sie mit `ifdown xxx; ifup xxx`, wobei Sie `xxx` durch den Schnittstellennamen ersetzen. Um die gesamte Konfiguration neu zu aktivieren, führen Sie `service network restart` aus.

## IPv6-Konfiguration

Wenn Sie auch IPv6 nutzen möchten, definieren Sie die betreffende Schnittstelle in `/etc/network/interfaces` einfach ein zweites Mal mit dem Schlüsselwort `inet6`. Das Schlüsselwort `auto` gibt an, dass die IPv6-Konfiguration das sogenannte *Router Advertisement* des Gateways bzw. IPv6-Routers berücksichtigt.

```
# IPv4-Konfiguration via DHCP, IPv6-Konfiguration via Router Advertisement
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 auto
```

Wenn das IPv6-Gateway einen DHCPv6-Server verwendet, lautet die korrekte Methode `dhcp`. Wenn außerdem die Router-Adresse per *Router Advertisement* konfiguriert werden soll, ist die Zusatzoption `accept_ra 1` erforderlich. Das ist beispielsweise der Fall, wenn Sie als DHCP-Server `dnsmasq` mit der Option `enable-ra` einsetzen.

```
# IPv4- und IPv6-Konfiguration via DHCP
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 dhcp
    accept_ra 1
```

Bei einer statischen Konfiguration muss `interfaces` so aussehen:

```
# IPv4- und IPv6-Konfiguration statisch
auto eth0
iface eth0 inet static
    ... (IPv4-Konfiguration wie bisher)
iface eth0 inet6 static
    address 2a01:4f8:161:107::2
    netmask 64
    gateway fe80::1
```

### WLAN- und WPA-Konfiguration

Um WLAN-Schnittstellen und die WPA-Authentifizierung kümmert sich auf Notebooks normalerweise der NetworkManager. Es besteht aber auch die Möglichkeit, die Konfiguration direkt in `/etc/interfaces` vorzunehmen. Besonders häufig ist dies auf Embedded-Geräten oder Minicomputern wie dem Raspberry Pi erforderlich.

Die Konfiguration der WLAN-Schnittstellen erfolgt grundsätzlich wie die von Ethernet-Schnittstellen. Dabei steuern Sie das Zusammenspiel mit dem Programm `wpa_supplicant` für die Authentifizierung durch `wpa`-Parameter. Im einfachsten Fall kann die Konfiguration so aussehen:

```
# /etc/network/interfaces (manuelle WLAN-Konfiguration mit WPA)
auto wlan0
iface wlan0 inet dhcp
    wpa-conf /etc/wpa.conf
```

Dabei gibt `wpa-conf` den Ort einer WPA-Konfigurationsdatei an, die wie das folgende Beispiel der Syntax von `wpa_supplicant` entsprechen muss:

```
# Datei /etc/wpa.conf
network={
    ssid="wlan-name"
    psk="ganzGeheim"
    proto=RSN
    key_mgmt=WPA-PSK
    pairwise=CCMP
    auth_alg=OPEN
}
```

Wenn es wie unter Raspbian eine grafische Benutzeroberfläche zur WLAN-Konfiguration gibt, um die Verbindungsdaten zu *mehreren* WLAN-Netzen zu verwalten, müssen Sie das Schlüsselwort `wpa-roam` verwenden:

```
# /etc/network/interfaces (WLAN-Defaultkonfiguration unter Raspbian)
allow-hotplug wlan0
iface wlan0 inet manual
    wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Die Zeile `allow-hotplug` erlaubt es anderen Programmen, die WLAN-Schnittstelle zu steuern. Die mit `wpa-roam` beginnende Zeile gibt an, in welcher Datei die WPA-Schlüssel verschiedener WLAN-Netze gespeichert sind. Sobald ein passendes Netz in Reichweite ist, wird automatisch eine Verbindung hergestellt, wobei die IP-Daten dann via DHCP bezogen werden. Die Anweisung `iface default inet dhcp` besagt, dass jedes in `wpa_supplicant.conf` definierte Netzwerk standardmäßig DHCP zur IP-Konfiguration verwenden soll.

Eine Menge weiterer `wpa`-Parameter für die `interfaces`-Datei sind in dieser Datei dokumentiert:

`/usr/share/doc/wpasupplicant/README.modes.gz`

### Netzwerkbrücken und Spezialeinstellungen

Zur Konfiguration virtueller Brücken oder zur Erfüllung spezieller Konfigurationswünsche gibt es eine Menge weiterer Schlüsselwörter, die auf der `man`-Seite zu `interfaces` dokumentiert sind. Insbesondere können Sie mit den Schlüsselwörtern `pre-up`, `up` und `post-up` sowie `pre-down`, `down` und `post-down` Kommandos angeben, die unmittelbar vor, während bzw. nach dem Einrichten bzw. Stoppen einer Netzwerkschnittstelle ausgeführt werden sollen. Die folgenden Zeilen zeigen eine Netzwerkbrücke, die mit der physischen Schnittstelle `eth0` verbunden ist und Netzwerkpakete zu vier virtuellen Maschinen weiterleitet:

```
# IPv4-Netzwerkbrücke
auto br0
iface br0 inet static
    # statische Konfiguration
    address    210.211.212.18
    broadcast  210.211.212.31
    netmask    255.255.255.224
    gateway    210.211.212.1
    pointopoint 210.211.212.1

# Brücke
bridge_ports eth0
bridge_stp off
bridge_fd 0
bridge_maxwait 0

# zusätzliche IPs für KVM
up route add -host 210.211.212.26 dev br0
up route add -host 210.211.212.27 dev br0
up route add -host 210.211.212.28 dev br0
up route add -host 210.211.212.29 dev br0
```

```
# statische Routen für IPv6
up ip -6 addr add 2a01:1234:567:890::2/64      dev br0
up ip route add default via fe80::1           dev br0
down ip route del default via fe80::1         dev br0
down ip -6 del add 2a01:1234:567:890::2/64    dev br0
```

### /etc/systemd/journald.conf

Der zu Systemd gehörende Protokolldienst *Journal* wird durch die folgenden Konfigurationsdateien gesteuert:

```
/etc/systemd/journald.conf
/etc/systemd/journald.conf.d/*.conf
/run/systemd/journald.conf.d/*.conf
[/usr]/lib/systemd/journald.conf.d/*.conf
```

Bei vielen Distributionen existiert ausschließlich `/etc/systemd/journald.conf`, und auch diese Datei enthält nur auskommentierte Beispielzeilen. In diesem Fall gelten die Defaulteinstellungen.

Zur Veränderung der Konfiguration können Sie an den oben aufgezählten Orten innerhalb der Konfigurationsgruppe `[Journal]` die folgenden Schlüsselwörter verwenden:

- ▶ **ForwardToSyslog**  
gibt an, ob durch das Journal protokollierte Nachrichten auch an einen traditionellen Syslog-Dienst weitergegeben werden sollen.
- ▶ **MaxFileSec**  
gibt an, nach welcher Zeit spätestens eine neue Logging-Datei gestartet werden soll. Die Defaulteinstellung sieht einen Monat vor. Diese Einstellung ist nur relevant, wenn die Logging-Dateien langsamer wachsen, als die Limits `SystemMaxUse`, `SystemMaxFileSize` und `SystemKeepFree` vorgeben.
- ▶ **MaxLevelStore**  
gibt die Prioritätsstufe an, bis zu der Nachrichten im Journal gespeichert werden. Die Defaulteinstellung lautet `debug`.
- ▶ **SystemKeepFree**  
gibt an, wie viel Prozent des Dateisystems frei bleiben müssen. Die Defaulteinstellung beträgt 15 Prozent.

► `SystemMaxFileSize`

gibt die maximale Größe einer Logging-Datei an. Die Defaulteinstellung beträgt ein Achtel von `SystemMaxUse`. Das führt zu einem automatischen »Rotating«, wobei neben der aktuellen Logging-Datei maximal sieben ältere Dateien entstehen.

► `SystemMaxUse`

gibt an, wie viel Prozent des Dateisystems die Logging-Dateien maximal beanspruchen können. Bevor dieses Limit überschritten wird, werden alte Logging-Dateien gelöscht. Die Defaulteinstellung beträgt 10 Prozent.

Unzählige weitere Optionen können Sie mit `man journald.conf` nachlesen.

## `/etc/locale.conf`

Die Datei `/etc/locale.conf` enthält bei Distributionen, die eine aktuelle Systemd-Version als Init-System verwenden, die Sprach- und Zeichensatzeinstellungen. Bei der Veränderung der Einstellungen hilft das Kommando `localectl`. Die folgende Aufzählung beschreibt die wichtigsten Parameter, die eingestellt werden können.

► `LANG=xxx`

bestimmt den Standardwert für alle nicht eingestellten LC-Variablen.

► `LC_ALL=xxx`

überschreibt alle individuellen LC-Einstellungen.

► `LC_COLLATE=xxx`

bestimmt die Sortierordnung.

► `LC_CTYPE=xxx`

bestimmt den Zeichensatz.

► `LC_MESSAGES=xxx`

bestimmt die Darstellung von Nachrichten, Fehlermeldungen etc.

► `LC_MONETARY=xxx`

bestimmt die Darstellung von Geldbeträgen.

► `LC_NUMERIC=xxx`

bestimmt die Darstellung von Zahlen.

► LC\_PAPER=xxx

bestimmt die Papiergröße.

► LC\_TIME=xxx

bestimmt die Darstellung von Datum und Uhrzeit.

Eine Liste aller möglichen Einstellungen liefert `locale -a`. Üblicherweise wird die Schreibweise `spr_land.zs` verwendet, wobei `spr` durch zwei Buchstaben die Sprache und `land` durch zwei Buchstaben das Land bezeichnet. `zs` gibt den Zeichensatz an. Im deutschen Sprachraum sollten Sie `de_DE.utf8` verwenden.

Je nach Distribution werden die Lokalisierungseinstellungen an anderen Orten gespeichert:

Debian, Ubuntu:	<code>/etc/default/locale</code>
RHEL 6:	<code>/etc/sysconfig/i18n</code>
SUSE:	<code>/etc/sysconfig/language</code>

### Beispiel

Mit der folgenden Einstellung in `locale.conf` gilt Deutsch als Defaultsprache und UTF-8 als Zeichensatz:

```
LANG="de_DE.utf8"
```

### `/etc/login.defs`

`/etc/login.defs` enthält diverse Einstellungen, die beim Anlegen neuer Benutzer und Gruppen berücksichtigt werden. Die Einstellungen gelten für die Kommandos aus dem `shadow-utils`-Paket, also z. B. für `adduser` und `useradd` sowie für `addgroup` und `groupadd`. Die `login.defs`-Parameter haben hingegen keinen Einfluss auf Kommandos, die auf die Pluggable Authentication Modules (PAM) zurückgreifen; dazu zählt insbesondere `passwd`!

Die Einstellungen in `login.defs` erfolgen zeilenweise. In jeder Zeile werden zuerst ein Schlüsselwort, dann ein oder mehrere Leer- oder Tabulatorzeichen und schließlich die gewünschte Einstellung angegeben.

► CREATE\_HOME

gibt an, ob beim Anlegen neuer Benutzer automatisch auch ein Heimatverzeichnis erzeugt werden soll (yes/no).



► `ENCRYPT_METHOD`

gibt an, welcher Algorithmus zur Speicherung der Hash-Codes in /etc/shadow verwendet werden soll. Zulässige Einstellungen sind DES, MD5, SHA256 und SHA512 (zurzeit am sichersten).

► `ENV_PATH` und `ENV_SUPATH`

enthält eine Voreinstellung für die PATH-Umgebungsvariable für gewöhnliche Benutzer und für root bzw. su. Weitere Einstellungen für die PATH-Variable erfolgen bei den meisten Distributionen in /etc/profile oder in /etc/bashrc.

► `GID_MIN` und `GID_MAX`

gibt die minimale und maximale Gruppen-ID-Nummer für gewöhnliche Gruppen an.

► `LOG_OK_LOGINS`

zeichnet erfolgreiche Logins mit Syslog auf (yes/no).

► `MAIL_DIR`

gibt den Ort des Verzeichnisses zur Speicherung lokaler E-Mails an.

► `PASS_MAX_DAYS`, `PASS_MIN_DAYS` und `PASS_WARN_AGE`

gibt an, wie lange Passwörter gelten und nach welchem Zeitraum sie frühestens verändert werden dürfen (siehe chage).

► `PASS_MIN_LEN`

definiert die minimale Passwortlänge.

► `SYS_GID_MIN` und `SYS_GID_MAX`

gibt die minimale und maximale GID für System-Accounts an.

► `SYS_UID_MIN` und `SYS_UID_MAX`

gibt die minimale und maximale UID für System-Accounts an.

► `SYSLOG_SU_ENAB` und `SYSLOG_SG_ENAB`

zeichnet su- und newgrp-Kommandos mit Syslog auf (yes/no).

► `UID_MIN` und `UID_MAX`

gibt die minimale und maximale Benutzer-ID-Nummer für gewöhnliche Benutzer an.

### ► USERGROUPS\_ENAB

steuert, ob zusammen mit jedem Benutzer eine eigene Gruppe erzeugt werden soll (yes/no).

### Beispiel

Die folgenden Zeilen zeigen die Defaulteinstellungen in `login.defs` unter Fedora. Beachten Sie aber, dass die Einstellungen für `PASS_MIN_LEN` und `UMASK` für den Betrieb von Fedora nicht relevant sind! Die Mindestanforderungen für Passwörter sind durch das PAM-Modul `pwquality` vorgegeben, und `umask` wird in `/etc/profile` mit 022 voreingestellt.

```
MAIL_DIR      /var/spool/mail
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_MIN_LEN  5
PASS_WARN_AGE 7
UID_MIN        1000
UID_MAX        60000
SYS_UID_MIN    201
SYS_UID_MAX    999
GID_MIN        1000
GID_MAX        60000
SYS_GID_MIN    201
SYS_GID_MAX    999
CREATE_HOME    yes
UMASK          077
USERGROUPS_ENAB yes
ENCRYPT_METHOD  SHA512
```

### `/etc/mdadm/mdadm.conf`

Die Datei `/etc/mdadm/mdadm.conf` fasst die Konfiguration des Software-RAID-Systems des Rechners zusammen. Wenn die Datei fehlt oder verloren gegangen ist, kann sie aus den in RAID-Partitionen gespeicherten Metadaten mit dem folgenden Kommando teilweise neu erstellt werden:

```
root# mdadm --examine --scan > /etc/mdadm/mdadm.conf
```

Die folgende Auflistung nennt die wichtigsten Schlüsselwörter in `mdadm.conf`:

### ► `ARRAY device1 metadata=... UUID=... name=... devices=...`

gibt an, aus welchen Komponenten ein RAID-Verbund besteht.

- ▶ `CREATE owner=... group=... mode=... auto=...`  
gibt an, welche Defaulteinstellungen beim Aktivieren eines RAID-Devices gelten sollen. Der `auto`-Parameter entspricht der Option `--auto` des Kommandos `mdadm`.
- ▶ `DEVICE device1 device2 | partitions | containers`  
gibt an, welche Festplatten oder Partitionen RAID-Komponenten enthalten können. Wenn die `DEVICE`-Angabe fehlt, gilt `DEVICE partitions containers`. Der Befehl `mdadm` durchsucht damit alle SSD- und Festplattenpartitionen sowie alle Logical Volumes.
- ▶ `MAILADDR name`  
gibt an, an welche lokale E-Mail-Adresse Meldungen, Warnungen und Fehlermeldungen gesendet werden sollen, sofern `mdadm` im Monitor-Modus läuft.

## Beispiel

Die folgenden Zeilen zeigen die Konfigurationsdatei eines Linux-Servers mit zwei RAID-1-Devices:

```
CREATE owner=root group=disk mode=0660 auto=yes
MAILADDR root
ARRAY /dev/md/0 metadata=1.2 UUID=0860...f0a2 name=rescue:0
ARRAY /dev/md/1 metadata=1.2 UUID=2546...e6e2 name=rescue:1
```

## /boot/grub/menu.lst

`/boot/grub/menu.lst` enthält die Konfiguration für den Linux-Bootloader GRUB in der veralteten Version 0.97. Diese GRUB-Version ist außer bei RHEL 6 nur noch bei älteren Linux-Installationen im Einsatz. Aktuelle Linux-Distributionen verwenden GRUB 2. Dessen Konfigurationsdatei heißt `/boot/grub/grub.cfg`.

Bei CentOS 6 bzw. RHEL 6 ist `/boot/grub/menu.lst` ein Link auf `/etc/grub/grub.conf`. Für die automatische Aktualisierung von `menu.lst` bei Kernel-Updates ist das Script `grub-by` verantwortlich. Direkte Veränderungen an `menu.lst` gehen deswegen bei Kernel-Updates verloren. Ähnlich geht Debian 6 vor, wo sich das Script `update-grub` um die Aktualisierung von `menu.lst` kümmert. `update-grub` berücksichtigt dabei in Kommentartform eingebettete Optionen innerhalb von `menu.lst`.

## Syntax

`menu.lst` beginnt mit allgemeinen Einstellungen, die unter anderem festlegen, welcher Menüeintrag standardmäßig aktiv ist, wie lange GRUB beim Rechnerstart auf interaktive Eingaben wartet etc. Danach folgen die Detailinformationen zu den

Menüeinträgen. Jeder Menüeintrag beginnt mit `title` eintrag, wobei eintrag der im GRUB-Menü angezeigte Text ist. Zum Start von Linux müssen anschließend zumindest drei Parameter eingestellt werden: der Ort der Systempartition mit `root`, der Ort der Kerneldatei mit `kernel` und der Ort der Initrd-Datei mit `initrd`.

Zum Verzweigen in den Windows-Bootloader geben Sie in `menu.lst` mit dem Schlüsselwort `rootnoverify` die Partition mit den Windows-Bootdaten an. Der Eintrag `chainloader +1` bewirkt, dass GRUB den Windows-Bootloader startet.

Für die Angabe von Partitionen gilt in GRUB eine eigene Nomenklatur in der Form `(h $d_n$ ,  $m$ )`. Dabei ist  $n$  die Nummer der Festplatte oder SSD,  $m$  die Nummer der Partition. Die Nummerierung beginnt in beiden Fällen mit 0.

### Beispiel

Eine minimale `menu.lst`-Datei zum wahlweisen Start von Linux oder Windows sieht so aus:

```
# globaler Bereich
default 0      # der erste Menüeintrag gilt als Standardeintrag
timeout 30     # 30 Sekunden warten, bevor das
               # Standardsystem gestartet wird

# Menüeintrag für Linux
title Linux
  root (hd0,3)
  kernel /boot/vmlinuz root=/dev/sda4
  initrd /boot/initrd

# Menüeintrag für Windows
title Windows
  rootnoverify (hd0,0)
  chainloader +1
```

### `/etc/modules`

Die Debian- und Ubuntu-spezifische Datei `/etc/modules` enthält zeilenweise die Namen von Kernelmodulen, die beim Rechnerstart geladen werden sollen. Normalerweise erkennt Linux selbst, welche Kernelmodule geladen werden sollen. Nur wenn das nicht erforderlich ist, müssen Sie der Datei `/etc/modules` die betreffenden Modulnamen hinzufügen.

## Alternative Verfahren

Kernelmodule, die bereits während des Bootprozesses erforderlich sind, müssen in die Initrd-Datei eingebaut werden. Dabei helfen die Kommandos `dracut` (Fedora, CentOS, RHEL) und `update-initramfs` (Debian, Ubuntu).

Um bei anderen Distributionen ein Kernelmodul am Ende des Init-Prozesses zu laden, fügen Sie die Anweisung `modprobe modulname` in die folgenden Dateien ein:

Red Hat, Fedora: `/etc/rc.d/rc.local`  
 SUSE: `/etc/init.d/boot.local`

Bei Fedora existiert `rc.local` standardmäßig nicht. Sie müssen die Datei erzeugen und ausführbar machen (`chmod a+x`).

Aktuelle Versionen von Raspbian verwenden Device Trees, um die zur Hardware passenden Kernelmodule zu verwalten. Die Konfiguration erfolgt durch `/boot/config.txt`. Die Device-Tree-Dateien mit der Kennung `*.dtb` befinden sich in den Verzeichnissen `/boot` und `/boot/overlays`.

### `/etc/sysconfig/network`

Die Red-Hat-spezifische Datei `/etc/sysconfig/network` enthält bei alten RHEL- und CentOS-Versionen Grundeinstellungen für die Netzwerkkonfiguration. Bei aktuellen Distributionen ist diese Datei optional, weil IPv6 standardmäßig aktiv ist und der Hostname nun in `/etc/hostname` eingestellt wird.

- ▶ `FORWARD_IPV4="yes|no"`  
 aktiviert IPv4-Forwarding. Der Parameter ist obsolet. Aktivieren Sie das Forwarding bei Bedarf in `/etc/sysctl.conf`.
- ▶ `GATEWAY="n.n.n.n"`  
 gibt die Adresse des Default-Gateways an. Die Gateway-Adresse kann auch schnittstellenspezifisch in `/etc/sysconfig/network-scripts/ifcfg-xxx` eingestellt werden.
- ▶ `HOSTNAME="name.domain"`  
 stellt den Hostnamen ein. Bei Distributionen mit einer aktuellen Systemd-Version ist dieser Parameter obsolet, der Hostname wird in `/etc/hostname` eingestellt.
- ▶ `NETWORKING="yes"`  
 aktiviert die Netzwerkfunktionen. Diese Einstellung gilt bei aktuellen Distributionen standardmäßig.

- `NETWORKING_IPV6="yes"`

aktiviert IPv6. Diese Einstellung gilt mittlerweile standardmäßig; sie ist nur bei älteren Distributionen erforderlich.

### `/etc/nsswitch.conf`

Die Datei `/etc/nsswitch.conf` steuert, wie Host-, Benutzer- und Gruppennamen durch die Name-Switch-Funktionen in der GNU-C-Bibliothek aufgelöst werden. Änderungen an dieser Datei sind selten erforderlich – es sei denn, im lokalen Netzwerk wird LDAP oder ein anderer Netzwerkdienst zur Benutzerverwaltung und -authentifizierung eingesetzt.

Die Datei enthält zeilenweise Einträge, wobei die erste Spalte jeweils den Namen einer Datenbank angibt. Mit »Datenbank« ist hierbei eine Gruppe von Informationen gemeint, also z. B. die Liste aller Benutzer samt Login-Namen, vollständigem Namen, Heimatverzeichnis, Default-Shell etc. Zulässige Schlüsselwörter für die erste Spalte sind unter anderem `aliases` (Mail-Aliases), `group`, `hosts`, `networks`, `passwd`, `rpc` und `shadow`.

Dem Datenbanknamen folgt ein Doppelpunkt und dann eine durch Leerzeichen getrennte Auflistung von Diensten. Die wichtigsten Dienste sind:

- `files` liest die Daten aus lokalen Konfigurationsdateien, z. B. `/etc/passwd` oder `/etc/group`.
- `compat` hat eine ähnliche Bedeutung wie `files`, erlaubt aber die Interpretation von Zusatzinformationen in `/etc/passwd`.
- `db` liest die Daten aus Datenbankdateien.
- `nis` wertet den veralteten Network Information Service aus.
- `dns` fragt beim Nameserver nach (für Hostnamen).
- `mdns4_minimal` verwendet Zeroconf zur Auflösung von Hostnamen.
- `ldap` kommuniziert mit einem LDAP-Server.
- `wins` greift auf einen WINS-Server zurück.

Außerdem können in der Form `[STATUS=ACTION]` bzw. `[!STATUS=ACTION]` Kommandos angegeben werden, die beim (Nicht-)Eintreten eines bestimmten Ereignisses ausgeführt werden sollen. Zulässige STATUS-Werte sind `success`, `notfound`, `unavail` und `tryagain`. Mögliche Aktionen sind `return` (Ergebnis sofort zurückgeben, nicht auf andere Optionen zurückgreifen) und `continue` (den nächsten Lookup-Dienst verwenden).

## Beispiel

Die folgenden Zeilen zeigen einen Auszug aus der Defaultkonfiguration unter Fedora. Für die Benutzeraccounts werden ausschließlich die Dateien `/etc/passwd`, `/etc/shadow` und `/etc/group` berücksichtigt. Hostnamen werden aus `/etc/hosts` gelesen, beim Nameserver nachgefragt sowie via Zeroconf (Avahi) ermittelt.

```
passwd:    files
shadow:    files
group:     files
hosts:     files dns mdns4_minimal
...
```

## `/etc/os-release`

Bei aktuellen Distributionen, die Systemd als Init-System verwenden, enthält `/etc/os-release` Informationen über die installierte Distribution. Die folgenden Zeilen zeigen die Syntax der Datei:

```
NAME="CentOS Linux"
VERSION="7 (Core)"
ID="centos"
ID_LIKE="rhel fedora"
VERSION_ID="7"
PRETTY_NAME="CentOS Linux 7 (Core)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:centos:centos:7"
HOME_URL="https://www.centos.org/"
BUG_REPORT_URL="https://bugs.centos.org/"
```

Bei älteren Distributionen bzw. bei Distributionen ohne Systemd werden diese Daten in anderen Dateien und mit anderer Syntax gespeichert:

```
Debian 7: /etc/lsb-release
RHEL 6:   /etc/redhat-release, /etc/system-release
```

## `/etc/passwd`

Die Datei `/etc/passwd` enthält Daten zu allen lokalen Accounts auf dem Rechner, darunter den Login-Namen, den vollständigen Namen, die Standard-Shell und das Heimatverzeichnis. *Nicht* enthalten ist hingegen der Hash-Code des Passworts – dieser ist aus Sicherheitsgründen in der separaten Datei `/etc/shadow` gespeichert. Zur Verwaltung der Accounts sollten Sie nach Möglichkeit die dafür vorgesehenen Kommandos verwenden, z. B. `useradd`, `usermod` oder `userdel`.

Die `passwd`-Datei besteht aus sieben Spalten, die jeweils durch Doppelpunkte voneinander getrennt sind:

- ▶ **Die erste Spalte (Account-Name)** enthält den Login-Namen. Dieser darf keine Leerzeichen enthalten und sollte frei von Sonderzeichen sein. Es ist üblich, ausschließlich Kleinbuchstaben zu verwenden.
- ▶ **Die zweite Spalte (Passwort)** enthielt in der Vergangenheit den Hash-Code des Passworts. Alle aktuellen Linux-Distributionen speichern in dieser Spalte nun das Zeichen `x`, die Hash-Codes befinden sich in `/etc/shadow`.
- ▶ **Die dritte Spalte (UID)** enthält die interne Benutzer-ID-Nummer, die für die Verwaltung der Zugriffsrechte und Prozesse verwendet wird. Bei den meisten Distributionen sind UIDs kleiner 1000 für System-Accounts reserviert und UIDs ab 1000 für reguläre Benutzer vorgesehen.
- ▶ **Die vierte Spalte (primäre Gruppe)** gibt die Nummer der primären Gruppe des Benutzers an. Der Benutzer kann weiteren Gruppen zugeordnet sein; sekundäre Gruppen werden in `/etc/group` gespeichert.
- ▶ **Die fünfte Spalte (Kommentar)** ist für eine Kommentar-Zeichenkette vorgesehen, die aus mehreren Teilen bestehen kann. In der Praxis wird hier zumeist nur der vollständige Benutzername gespeichert. Wenn dieser beim Anlegen des Benutzers nicht angegeben wurde, stimmt die fünfte Spalte mit der ersten überein.
- ▶ **Die sechste Spalte (Heimatverzeichnis)** enthält den Pfad zum Heimatverzeichnis.
- ▶ **Die siebte Spalte (Shell)** gibt die Shell an, die nach dem Login automatisch aktiv ist – bei gewöhnlichen Benutzern oft `/bin/bash`. Bei System-Accounts, für die kein Login vorgesehen ist, wird statt der Shell oft `/sbin/nologin` oder `/bin/false` verwendet. Die Default-Shell kann mit dem Kommando `chsh` verändert werden.

### Beispiel

Die folgenden Zeilen zeigen einige System-Accounts sowie einen Benutzer-Account auf einem Fedora-System:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
...
kofler:x:1000:1000:Michael Kofler:/home/kofler:/bin/bash
```



## /etc/profile

/etc/profile sowie /etc/profile.d/\*.sh enthalten systemweite Einstellungen von Umgebungsvariablen. Dort werden Variablen wie PATH, MAIL, HOSTNAME voreingestellt. Außerdem wird hier bei vielen Distributionen mit umask eingestellt, welche Zugriffsbits bei neu erzeugten Dateien gesetzt werden sollen.

/etc/profile ist eine Shell-Script-Datei. Syntaktisch ist somit alles erlaubt, was der gemeinsame Nenner der wichtigsten Shells zulässt. Unter Ubuntu ist ergänzend zu /etc/profile die Datei /etc/environment für die systemweite Einstellung von Umgebungsvariablen vorgesehen. Tatsächlich enthält diese Datei üblicherweise aber nur die Einstellung der PATH-Variablen.

Die profile-Einstellungen werden durch benutzerspezifische Einstellungen in .profile sowie durch Shell-spezifische Einstellungen ergänzt. Die bash liest z. B. die Dateien /etc/bashrc, /etc/bash.bashrc, .bashrc und .alias. Je nach Distribution kann es weitere Konfigurationsdateien geben. Beispielsweise wird PATH in Debian und Ubuntu durch /etc/environment eingestellt. Für die im Grafiksystem X gültigen Umgebungsvariablen werden auch die Dateien des Verzeichnisses /etc/X11/Xsession.d/ verarbeitet, falls dieses existiert.

## /etc/resolv.conf

/etc/resolv.conf steuert, wie die IP-Adressen für unbekannte Netzwerknamen (Hostnamen) ermittelt werden. »Unbekannt« bedeutet, dass die Namen nicht in hosts definiert sind. Die Datei besteht normalerweise nur aus zwei oder drei Zeilen mit den folgenden Schlüsselwörtern:

### ► nameserver *ipaddr*

gibt die IP-Adresse eines Nameservers an. Das Schlüsselwort kann mehrfach verwendet werden, um alternative Nameserver anzugeben. Insgesamt sind maximal je drei IPv4- und IPv6-Nameserver erlaubt.

### ► domain *mydomain*

gibt den lokalen Domainnamen an. Das ermöglicht die verkürzte Angabe von Hostnamen, also von name anstelle von name.mydomain.

### ► search *domain*

gibt einen Domainnamen an, der bei Suchanfragen berücksichtigt wird. Insgesamt können bis zu sechs Domainnamen angegeben werden, jeweils durch einen eigenen search-Eintrag. Bei der Namensauflösung werden dann der Reihe nach die angegebenen Domainnamen getestet, bei ping name also zuerst name, dann

`name.domain1`, dann `name.domain2` etc. Die Defaulteinstellung für `search` ist der lokale Domainname.

`domain` und `search` schließen sich gegenseitig aus. Wenn `resolv.conf` beide Schlüsselwörter enthält, gilt das zuletzt genannte.

`/etc/resolv.conf` wird häufig dynamisch erzeugt, insbesondere dann, wenn die IP-Konfiguration über DHCP erfolgt. Ubuntu richtet ab Version 12.04 standardmäßig einen lokalen Nameserver ein. Dabei kommt das Programm `Dnsmasq` zum Einsatz. Die Nameserver-Adresse in `/etc/resolv.conf` lautet deswegen `127.0.0.1`. Wenn Sie den Nameserver manuell einrichten möchten, müssen Sie dessen Adresse in `/etc/network/interfaces` mit dem Schlüsselwort `dns-nameservers` angeben.

### Beispiel

```
# /etc/resolv.conf
domain mylan                # Domainname des LANs
nameserver 211.212.213.1    # IPv4-DNS
nameserver 211.212.214.1    # Ersatz-DNS (falls der erste ausfällt)
nameserver 2001:4860:4860::8888 # IPv6-DNS
```

### `/etc/rsyslog.conf`

Viele Linux-Distributionen verwenden für die Protokollierung der Meldungen des Kernels sowie diverser Netzwerkdienste das Syslog-kompatible Programm `rsyslogd`. Die Konfiguration dieses Programms erfolgt durch die Datei `/etc/rsyslog.conf` sowie die Ergänzungsdateien `/etc/rsyslog.d/*.conf`. Diese Konfigurationsdateien enthalten Regeln, die aus zwei Teilen bestehen:

- **Selektor:** Der erste Teil jeder Regel gibt an, was protokolliert werden soll.
- **Aktion:** Der zweite Teil steuert, was mit der Meldung geschehen soll.

Regeln können mit dem Zeichen `\` über mehrere Zeilen verteilt werden. Es ist möglich, dass auf eine Meldung mehrere Regeln zutreffen. In diesem Fall wird die Meldung mehrfach protokolliert bzw. weitergegeben. Damit Änderungen an der Syslog-Konfiguration wirksam werden, muss der Syslog-Dienst mit `service rsyslog restart` neu gestartet werden!

### Selektoren

Jeder Selektor besteht aus zwei durch einen Punkt getrennten Teilen: *dienst.prioritätsstufe*. Es ist erlaubt, mehrere durch einen Strichpunkt separierte Selektoren anzugeben. Des Weiteren können in *einem* Selektor mehrere Dienste durch Kommas

getrennt werden. Alle Linux-Programme, die Syslog verwenden, müssen ihren Meldungen einen Dienst und eine Priorität zuordnen.

Syslog kennt die folgenden Dienste (Facilities): auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, syslog, user, uucp sowie local0 bis local7. Das Zeichen \* umfasst alle Dienste.

Syslog kennt außerdem diese Prioritätsstufen (in steigender Wichtigkeit): debug, info, notice, warning = warn, err = error, crit, alert und emerg = panic. Die Schlüsselwörter warn, error und panic gelten als veraltet – verwenden Sie stattdessen warning, err und emerg. Das Zeichen \* umfasst alle Prioritätsstufen. Das Schlüsselwort none gilt für Nachrichten, denen keine Priorität zugeordnet ist.

Die Angabe einer Prioritätsstufe schließt alle höheren (wichtigeren) Prioritätsstufen mit ein. Der Selektor mail.err umfasst also auch crit-, alert- und emerg-Meldungen des Mail-Systems. Wenn Sie explizit nur Nachrichten einer bestimmten Priorität wünschen, stellen Sie das Zeichen = voran (also etwa mail.=err).

## Aktionen

Als Aktion wird normalerweise der Name einer Logging-Datei angegeben. Normalerweise werden Logging-Dateien nach jeder Ausgabe synchronisiert. Wenn dem Dateinamen ein Minuszeichen vorangestellt ist, verzichtet Syslog auf die Synchronisierung. Das ist wesentlich effizienter, allerdings gehen dann bei einem Absturz noch nicht physikalisch gespeicherte Meldungen verloren.

Syslog kann Nachrichten auch an FIFO-Dateien (First In, First Out) oder Pipes weiterleiten. In diesem Fall stellen Sie dem Dateinamen das Zeichen | voran. Die Datei /dev/xconsole, die im folgenden Listing vorkommt, ist eine besondere FIFO-Datei zur Weitergabe von Meldungen an das Grafiksystem X.

Das Zeichen \* bedeutet, dass die Nachricht an alle in Konsolen bzw. via SSH eingeloggteten Benutzer gesendet wird. Da das sehr störend ist, wird es standardmäßig nur für kritische Meldungen verwendet.

## Journal-Konfiguration

Aktuelle Distributionen verwenden statt oder ergänzend zu rsyslogd das zum Systemd-Projekt gehörige Programm *Journal*. Dessen Konfiguration erfolgt durch /etc/systemd/journal.conf sowie durch einige weitere Dateien.

## Beispiel

Die folgenden Zeilen geben die Syslog-Standardkonfiguration von Ubuntu leicht gekürzt und etwas übersichtlicher formatiert wieder:

```
# Datei /etc/rsyslog.d/50-default.conf bei Ubuntu
# Selektor                                Aktion
auth,authpriv.*                          /var/log/auth.log
*.*;auth,authpriv.none                   -/var/log/syslog
kern.*                                    -/var/log/kern.log

mail.*                                    -/var/log/mail.log
mail.err                                  /var/log/mail.err
*.emerg                                   :omusrmsg:*

daemon.*;mail.*;\
    news.err;\
    *.=debug;*.=info;\
    *.=notice;*.=warn                    | /dev/xconsole
```

Im Klartext bedeutet die obige Konfiguration:

- ▶ `/var/log/auth` enthält Authentifizierungsmeldungen aller Prioritätsstufen. Dazu zählen gescheiterte und erfolgreiche Login-Versuche (auch via SSH), PAM-Meldungen, sudo-Kommandos etc. Als einzige Logging-Datei wird `auth` bei jeder Meldung sofort synchronisiert.
- ▶ `/var/log/syslog` enthält *alle* via Syslog protokollierten Meldungen (inklusive Authentifizierungsmeldungen, denen keine Priorität zugewiesen ist). Der allumfassende Ansatz ist zugleich ein Vor- und ein Nachteil: Einerseits können Sie so aus einer einzigen Datei alle erdenklichen Informationen extrahieren; andererseits ist es in diesem Sammelsurium natürlich besonders schwierig, relevante Einträge zu finden.
- ▶ `/var/log/kern.log` enthält alle Kernelmeldungen.
- ▶ Die Nachrichten des Mail-Systems (z. B. Postfix) werden über mehrere Dateien verteilt. In `mail.log` werden *alle* Nachrichten gespeichert, in `mail.err` nur Fehlermeldungen.
- ▶ Kritische Systemmeldungen, z. B. über einen bevorstehenden Shutdown oder über Kernelfehler, werden durch `:omusrmsg:*` an alle Benutzer weitergeleitet, genau genommen an alle Terminalfenster und Konsolen. `omusrmsg` ist ein `rsyslog`-Modul, um Nachrichten an Benutzer zu senden.
- ▶ Außerdem werden diverse Warnungen und Fehlermeldungen an das X-System weitergeleitet. Um diese Meldungen unter X zu verfolgen, starten Sie das Programm `xconsole`. Es sieht wie ein kleines Terminalfenster aus, erlaubt aber keine Eingaben.

## **/etc/services**

Die Datei `/etc/services` enthält eine Liste aller üblichen Netzwerkdienste und der ihnen zugeordneten IP-Ports und Protokolltypen. Veränderungen an dieser Datei sind nur in Ausnahmefällen erforderlich. Die Syntax der Datei geht aus den folgenden Beispielzeilen klar hervor:

```
# service-name port/protocol [aliases] [# comment]
tcpmux        1/tcp                # TCP port service multiplexer
tcpmux        1/udp                # TCP port service multiplexer
rje           5/tcp                # Remote Job Entry
rje           5/udp                # Remote Job Entry
echo          7/tcp
echo          7/udp
...
ftp           21/tcp
ftp           21/udp              fsp fspd
ssh           22/tcp              # The Secure Shell (SSH) Protocol
ssh           22/udp              # The Secure Shell (SSH) Protocol
telnet        23/tcp
telnet        23/udp
...
```

## **/etc/shadow**

`/etc/shadow` ergänzt die Benutzerinformationen aus `/etc/passwd` um den Passwort-Hash sowie um Daten, die angeben, wie lange der Account gültig ist bzw. wann das Passwort erneuert werden soll. Zur Veränderung dieser Daten verwenden Sie üblicherweise die Kommandos `passwd` und `chage`. Die `shadow`-Datei besteht aus neun Spalten, die durch Doppelpunkte getrennt sind:

- **Die erste Spalte (Account-Name)** stimmt mit der ersten Spalte von `/etc/passwd` überein.
- **Die zweite Spalte (Passwort-Hash)** ermöglicht die Überprüfung des Passworts. Eine Rekonstruktion des Passworts aus dem Hash-Code ist hingegen unmöglich. Bei aktuellen Linux-Distributionen wird der Hash-Code mit `sha512sum` erzeugt. Da beim Erzeugen jedes Hash-Codes ein zufälliger Initialisierungswert verwendet wird (das sogenannte *salt*), haben zwei Benutzer, die dasselbe Passwort verwenden, dennoch unterschiedliche Hash-Codes. Damit sind auch Wörterbuchangriffe unmöglich, bei denen vorgenerierte Hash-Codes mit dem Inhalt von `/etc/shadow` verglichen werden.

Bei Accounts, die keinen Login vorsehen, enthält die zweite Spalte einfach einen Stern. Bei gesperrten Accounts werden dem Hashcode zwei Ausrufezeichen vorangestellt. Der Code wird damit ungültig. Zur Reaktivierung des Accounts werden die Ausrufezeichen wieder entfernt.

- ▶ **Die dritte Spalte (letzte Änderung)** gibt an, wann das Passwort zuletzt verändert wurde. Dabei wird in Tagen ab dem 1.1.1970 gerechnet. Der Wert 0 bedeutet, dass der Benutzer das Passwort sofort beim nächsten Login ändern muss.
- ▶ **Die vierte Spalte (Mindestalter)** gibt an, nach wie vielen Tagen ein Passwort frühestens verändert werden darf. 0 erlaubt eine jederzeitige Änderung.
- ▶ **Die fünfte Spalte (Höchstalter)** gibt an, nach wie vielen Tagen ein Passwort spätestens verändert werden muss. 0 erlaubt eine unbeschränkte Nutzung des Passworts.
- ▶ **Die sechste Spalte (Warnzeit)** gibt an, wie viele Tage vor dem Ablauf des Passworts der Benutzer darauf hingewiesen wird. 0 deaktiviert diese Warnungen.
- ▶ **Die siebte Spalte (Sperrzeit)** gibt an, wie viele Tage nach Ablauf des Passworts das Konto gesperrt wird. Es kann dann nur noch vom Administrator durch passwd-u reaktiviert werden.
- ▶ **Die achte Spalte (Ablaufzeit)** gibt an, wann der Account abläuft. Der Zeitpunkt wird in Tagen ab dem 1.1.1970 angegeben. Wenn der Account unbegrenzt gelten soll, bleibt diese Spalte leer. Der Wert 0 wird nicht verwendet!
- ▶ **Die neunte Spalte** ist für zukünftige Erweiterungen reserviert.

### Beispiel

Die folgenden Zeilen zeigen einen Auszug aus `/etc/shadow`, wobei die langen SHA512-Hash-Codes gekürzt wurden:

```
root:$6$s18/.eHvacN2aUgr$y...FA1:15864:0:99999:7:::
bin:*:15813:0:99999:7:::
daemon:*:15813:0:99999:7:::
adm:*:15813:0:99999:7:::
..
kofler:$6$5bU6G.TOHSY1DxNn...mL.:15864:0:99999:7:::
```

Um herauszufinden, wann das Passwort zuletzt verändert wurde, führen Sie date aus:

```
user$ date -d "1970-01-01 + 15864 days"
Sam Jun  8 00:00:00 CEST 2013
```

Die weiteren chage-Angaben für die Benutzer root und kofler bedeuten, dass das Passwort jederzeit verändert werden kann (0) und nahezu unbegrenzt gültig ist (99999 = ca. 273 Jahre). Sieben Tage vor dem Ablauf des Passworts wird der Benutzer beim Login gewarnt.

## /etc/apt/sources.list

In `/etc/apt/sources.list` sowie `/etc/apt/sources.list.d/*` sind die APT-Paketquellen definiert. Die Syntax jeder Zeile sieht so aus:

```
pakettyp uri distribution [komponente1] [komponente2] [komponente3] ...
```

Der Pakettyp lautet `deb` für gewöhnliche Debian-Pakete bzw. `deb-src` für Quellcodepakete. Die zweite Spalte gibt das Basisverzeichnis der Paketquelle an. Neben HTTP- und FTP-Verzeichnissen unterstützt APT auch gewöhnliche Verzeichnisse, RSH- oder SSH-Server sowie CDs bzw. DVDs.

Die dritte Spalte bezeichnet die Distribution. Alle weiteren Spalten geben die Komponenten der Distribution an, die berücksichtigt werden können. Die Komponentennamen sind von der Distribution und von der Paketquelle abhängig! Beispielsweise unterscheidet Ubuntu zwischen *main*-, *restricted*-, *universe*- und *multiverse*-Paketen, während Debian zwischen den Komponenten *main*, *contrib*, *non-free* etc. differenziert.

Die zuerst genannten Paketquellen werden bevorzugt: Wenn ein bestimmtes Paket also in mehreren Quellen zum Download zur Verfügung steht, lädt APT es von der ersten Quelle herunter.

### Beispiel

Das folgende Listing zeigt einige Ubuntu-Paketquellen. Aus Platzgründen wurde dabei jeder Eintrag über zwei Zeilen verteilt.

```
# Datei /etc/apt/sources.list
deb http://de.archive.ubuntu.com/ubuntu/ wily          \
                                main restricted universe multiverse
deb http://de.archive.ubuntu.com/ubuntu/ wily-updates \
                                main restricted universe multiverse
deb http://security.ubuntu.com/ubuntu    wily-security \
                                main restricted universe multiverse
```

## /etc/sudoers

Die Datei `/etc/sudoers` sowie die Zusatzdateien in `/etc/sudoers.d` legen fest, welche Benutzer welche Kommandos mit welchen Rechten durch `sudo` ausführen dürfen. Die Datei enthält außerdem diverse allgemeine Einstellungen, die das Grundverhalten von `sudo` steuern. Die Einstellungen fallen je nach Distribution recht unterschiedlich aus.

Die `sudoers`-Datei sollte nur mit dem Kommando `visudo` verändert werden. Es ruft den durch die Umgebungsvariablen `VISUAL` oder `EDITOR` definierten Editor auf (standard-

mäßig vi), führt vor dem Speichern einen Syntaxcheck durch und stellt so sicher, dass Sie sich nicht durch eine fehlerhafte `sudoers`-Datei selbst von weiteren Administrationsarbeiten ausschließen. Besonders wichtig ist das bei Distributionen wie Ubuntu, die keinen `root`-Login vorsehen.

### Grundeinstellungen

- ▶ `Defaults always_set_home`  
verändert beim Benutzerwechsel die `HOME`-Umgebungsvariable, sodass diese auf das Heimatverzeichnis des neuen Benutzers verweist. Normalerweise geschieht dies automatisch. Diese Option ist nur dann relevant, wenn `!env_reset` gilt oder `env_keep` die Variable `PATH` enthält.
- ▶ `Defaults env_keep="var1 var2 var3"`  
gibt an, welche Umgebungsvariablen beim Benutzerwechsel erhalten bleiben sollen und somit von `env_reset` ausgenommen sind.
- ▶ `Defaults env_reset`  
bewirkt, dass beim Benutzerwechsel alle Umgebungsvariablen zurückgesetzt werden. Diese Einstellung gilt standardmäßig. Um sie zu deaktivieren, geben Sie `Defaults !env_reset` an.
- ▶ `Defaults mail_badpass`  
führt dazu, dass nach einem fehlerhaften Login-Versuch eine Warn-E-Mail an den Administrator versandt wird.
- ▶ `Defaults secure_path="pfad1:pfad2:pfad3"`  
legt den Inhalt der `PATH`-Umgebungsvariable für `sudo`-Kommandos fest.
- ▶ `Defaults targetpw`  
bedeutet, dass grundsätzlich das Passwort für den Account angegeben werden muss, in dem das Kommando ausgeführt werden soll, in der Regel also das `root`-Passwort. Ohne diese Einstellung erwartet `sudo` das Passwort des aktuellen Benutzers.
- ▶ `Defaults timestamp_timeout=n`  
gibt an, nach welcher Zeit `sudo` neuerlich nach dem Passwort fragt. Die Standardeinstellung beträgt fünf Minuten.



## Rechte

Dreispaltige Einträge in `/etc/sudoers` legen fest, welche Benutzer von welchem Rechner aus welche Programme ausführen dürfen. Die folgende Zeile bedeutet, dass die Benutzerin `kathrin` auf dem Rechner `uranus` das Kommando `/sbin/fdisk` ausführen darf. Das Schlüsselwort `ALL` bedeutet, dass `kathrin` das Kommando unter jedem beliebigen Account ausführen darf, also als `root`, als `news`, als `lp` etc.

```
kathrin uranus=(ALL) /sbin/fdisk
```

Wenn der ersten Spalte von `sudoers` das Zeichen `%` vorangestellt wird, gilt der Eintrag für alle Mitglieder der angegebenen Gruppe. Die folgende Zeile bedeutet, dass alle Benutzer der Gruppe `wheel` von jedem beliebigen Rechner aus alle Kommandos als beliebiger Benutzer ausführen dürfen:

```
%wheel ALL=(ALL) ALL
```

Es besteht die Möglichkeit, einem bestimmten Benutzer das Ausführen von `sudo` ohne Passwortangabe zu erlauben. Das `NOPASSWD`-Schlüsselwort ist allerdings nur gültig, wenn es keine anderen `sudoers`-Zeilen gibt, die vom selben Benutzer ein Passwort verlangen. Das gilt auch für Gruppeneinträge, also z. B. `%admin`.

```
kofler ALL=(ALL) NOPASSWD: ALL
```

Äußerst liberal ist die folgende Regel, die sinngemäß lautet: Jeder darf alles. Sinnvoll ist diese Regel nur in Kombination mit der Option `Defaults targetpw`, die grundsätzlich die Eingabe des Passworts des Benutzers verlangt, in dessen Namen ein Kommando ausgeführt werden soll.

```
ALL ALL=(ALL) ALL
```

## Beispiel

Die folgenden Zeilen zeigen die Defaultkonfiguration unter Ubuntu. Bei dieser Distribution ist `root` ohne gültiges Passwort eingerichtet. Ein `root`-Login ist damit unmöglich. Auch `su` oder `ssh -l root` funktionieren nicht. Die einzige Möglichkeit zur Ausführung administrativer Kommandos bietet somit `sudo`. Dieses Recht haben alle Mitglieder der Gruppen `admin` und `sudo`, wobei die `admin`-Gruppe nur noch aus Gründen der Kompatibilität zu älteren Ubuntu-Versionen enthalten ist. Bei aktuellen Ubuntu-Versionen und bei den meisten anderen Distributionen ist `sudo` die übliche Gruppe für Benutzer mit `sudo`-Rechten.

```
# Defaultkonfiguration in /etc/sudoers bei Ubuntu
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path=\
            "/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
```

```
root    ALL=(ALL:ALL) ALL
%admin  ALL=(ALL) ALL
%sudo   ALL=(ALL:ALL) ALL
```

### /etc/sysctl.conf

/etc/sysctl.conf enthält Defaulteinstellungen für Kernelparameter. Die Datei wird beim Systemstart ausgewertet, und alle dort aufgezählten Parameter werden entsprechend eingestellt. Bei nachträglichen Änderungen erreichen Sie mit `sysctl -p`, dass alle Einstellungen aus `sysctl.conf` ausgelesen und gesetzt werden.

Die einzustellenden Kernelparameter werden zeilenweise in der Form `name.name.name=wert` angegeben. Kommentarzeilen beginnen mit `;` oder mit `#`.

### Beispiel

Die folgende Einstellung in `sysctl.conf` bewirkt, dass beim Rechnerstart das Forwarding für IPv4 und für IPv6 aktiviert wird. Das ist erforderlich, wenn der Rechner als Router arbeiten soll.

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

### systemd.service

Das Init-System Systemd wird durch eine ganze Palette von Konfigurationsdateien gesteuert, die sich in den drei folgenden Verzeichnissen befinden:

```
/etc/systemd/system
/run/systemd/system
[/usr]/lib/systemd/system
```

Wenn Systemd auch zur Verwaltung benutzerspezifischer Dienste verwendet wird, gibt es auch `*.service`-Dateien in `/home/name/.config/systemd/user` sowie in einigen weiteren Verzeichnissen.

Die in diesem Abschnitt beschriebenen `*.service`-Dateien legen fest, wie ein Dienst – beispielsweise ein SSH-Server – durch Systemd gestartet, überwacht und bei Bedarf auch wieder gestoppt wird. `*.service`-Dateien bestehen aus mehreren Abschnitten, die durch Zeilen mit dem Inhalt `[Unit]`, `[Service]`, `[Install]` eingeleitet werden.

## [Unit]-Schlüsselwörter

- ▶ `Before/After=name`  
nennt andere Dienste bzw. Units, die vorher oder nachher gestartet werden sollen. Before und After geben Systemd Hinweise, in welcher Reihenfolge die Dienste gestartet werden sollen.
- ▶ `Description=beschreibung`  
beschreibt den Dienst durch eine Zeichenkette.
- ▶ `OnFailure=name`  
nennt einen oder mehrere Dienste, die ausgeführt werden sollen, wenn der Start des aktuellen Dienstes scheitert.
- ▶ `Wants/Requires=name`  
nennt andere Dienste, die vorher gestartet werden sollen bzw. müssen. Diese Schlüsselwörter definieren Abhängigkeiten. Systemd versucht, die betreffenden Dienste vorher zu starten. Bei Wants wird ein eventuelles Scheitern stillschweigend ignoriert, während es bei Requires dazu führt, dass der aktuelle Dienst nicht gestartet und stattdessen ein Fehler ausgelöst wird.

## [Service]-Schlüsselwörter

- ▶ `Environment=var1='wert1' var2='wert2' ...`  
definiert mehrere Umgebungsvariablen, die an den zu startenden Prozess weitergegeben werden.
- ▶ `EnvironmentFile=dateiname`  
liest die Datei und berücksichtigt die dort zeilenweise enthaltenen Variablenzuweisungen.
- ▶ `ExecReload=/pfad/programm optionen`  
führt das angegebene Kommando aus, um die Konfiguration des Dienstes im laufenden Betrieb neu zu laden.
- ▶ `ExecStart=/pfad/programm optionen`  
führt das angegebene Kommando aus, um den Dienst zu starten. In Kombination mit `Type=oneshot` sind mehrere ExecStart-Zeilen erlaubt, die dann der Reihe nach ausgeführt werden.

► `ExecStop=/pfad/programm optionen`

führt das angegebene Kommando aus, um den Dienst zu beenden. Wenn Sie dieses Schlüsselwort nicht verwenden, wird der Dienst durch ein KILL-Signal beendet. Welches Signal wie versendet werden soll, beschreiben bei Bedarf die Schlüsselwörter `KillMode` und `KillSignal`, die in der man-Seite `systemd.kill` beschrieben sind. `ExecStop` ist nur in Kombination mit `Type=oneshot` möglich.

► `RemainAfterExit`

wird oft in Kombination mit `Type=oneshot` verwendet. `Systemd` merkt sich den gerade aktivierten Zustand.

Ohne diese Option glaubt `Systemd` nach `systemctl start name`, dass die Aktion mit dem Ende des letzten `StartExec`-Kommandos beendet ist, und setzt den Status sofort wieder auf `stop`. Ein explizites Ausführen von `systemctl stop` würde dann wirkungslos bleiben.

► `Type=simple|forking|oneshot|dbus|notify|idle`

gibt an, um welche Art von Dienst es sich handelt bzw. wie das Programm gestartet werden soll.

Wenn die Typangabe fehlt, gilt `simple`. `Systemd` nimmt dann an, dass das mit `ExecStart` angegebene Kommando einen Hintergrunddienst startet. Erst wenn dieses Kommando endet, betrachtet `Systemd` den Dienst als regulär beendet.

`Type=notify` funktioniert so ähnlich, allerdings erwartet `Systemd` eine explizite Benachrichtigung darüber, dass der Startprozess abgeschlossen ist. Der Hintergrunddienst muss dazu die Funktion `sd_notify` oder eine gleichwertige Funktion aufrufen.

Auch `Type=forking` hat Ähnlichkeiten zu `simple`, allerdings ist hier das `ExecStart`-Kommando dafür verantwortlich, einen vom Startkommando losgelösten Hintergrundprozess zu starten. Sobald das `ExecStart`-Kommando abgeschlossen ist, nimmt `Systemd` an, dass der Dienst in einem eigenen Hintergrundprozess läuft.

Beim Typ `oneshot` nimmt `Systemd` an, dass das auszuführende Kommando so rasch beendet wird, dass darauf gewartet werden kann, bevor andere Aktionen durchgeführt werden. Der Status des Dienstes wird mit dem Ende wieder auf `stop` gesetzt, es sei denn, die Service-Datei enthält das Schlüsselwort `RemainAfterExit`.

Details zu den weiteren Type-Variante entnehmen Sie bitte man `systemd.service`.

## [Install]-Schlüsselwörter

- `WantedBy=` bzw. `RequiredBy=target`

gibt an, für welches Target der Dienst wünschenswert oder erforderlich ist, z. B. für das `multi-user.target` oder für das `reboot.target`. In der Praxis wird zumeist `WantedBy` eingesetzt. Das führt dazu, dass das Target auch dann erreicht wird, wenn der Start einzelner Dienste scheitert. `RequiredBy` führt bei Problemen hingegen zu einem Fehler. Das betreffende Target kann dann nicht aktiviert werden.

## Beispiel

Die folgenden Zeilen zeigen die Datei `httpd.service` in der Defaultkonfiguration von Fedora 23. Sie ist für den Start des Webservers Apache verantwortlich. Voraussetzungen für den Start sind die erfolgreiche Aktivierung des Netzwerks und eventueller Netzwerkdateisysteme. Der Webserver soll automatisch im Rahmen des Multi-User-Targets gestartet werden.

```
# Datei /lib/systemd/system/httpd.service (Fedora 23)
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target

[Service]
Type=notify
Environment=LANG=C
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
ExecReload=/usr/sbin/httpd $OPTIONS -k graceful
# Send SIGWINCH for graceful stop
KillSignal=SIGWINCH
KillMode=mixed
PrivateTmp=true

[Install]
WantedBy=multi-user.target
```

## Weitere Systemd-Konfigurationsdateien

Neben den `*.service`-Dateien gibt es zur Systemd-Konfiguration auch Dateien mit den Kennungen `*.mount`, `*.path`, `*.socket`, `*.target` und `*.wants`, auf die ich in diesem Buch nicht näher eingehe. Ihre Syntax ist in den `man`-Seiten `service.mount`, `service.path` etc. dokumentiert. Alle diese Dateien zusammen gelten als Unit-Dateien. Die allgemeine Syntax derartiger Units erläutert `man systemd.unit`.

Wenn Sie Jobs einrichten wollen, die durch Systemd periodisch ausgeführt werden, müssen Sie sich dem Format der `*.timer`-Dateien auseinandersetzen – siehe [systemd.timer](#).

### systemd.timer

Systemd kann sich ähnlich wie Cron auch um die regelmäßige Ausführung von Prozessen kümmern. Die Konfiguration erfolgt durch eine `name.timer`-Datei, der eine gleichnamige Service-Datei gegenüberstehen muss (also `name.service`). Bei selbst definierten Jobs befinden sich beide Dateien üblicherweise im Verzeichnis `/etc/systemd/system`.

Der Aufbau einer `*.timer`-Datei ist ähnlich wie der einer `*.service`-Datei (siehe `systemd.service`). Er besteht üblicherweise aus drei Abschnitten: den schon bekannten `[Unit]`- und `[Install]`-Abschnitten und einem `[Timer]`-Abschnitt, dessen Einstellungen steuern, wann bzw. wie häufig der Job ausgeführt wird.

#### [Timer]-Schlüsselwörter

► `AccuracySec=zeitspanne`

gibt an, innerhalb welcher Genauigkeit die Jobs ausgeführt werden sollen. Standardmäßig ist dafür ein Zeitfenster von einer Minute vorgesehen, innerhalb dessen die Startzeit zufällig bestimmt wird. Kleinere Zeitspannen sind möglich, sollten aber nur dann verwendet werden, wenn dies wirklich erforderlich ist.

► `OnBootSec=zeitspanne`

führt den Job erstmals nach der angegebenen Zeitspanne nach dem Bootprozess aus. Wenn die Zeitangabe ohne Einheit erfolgt, sind Sekunden gemeint. Die Syntax der Zeitangaben ist in `man systemd.time` dokumentiert. Zulässig ist z. B. `2h 15min` oder `2weeks` oder `4months`.

► `OnCalendar=zeitangabe`

gibt an, zu welcher Zeit der Job ausgeführt werden soll. `12:30` bedeutet beispielsweise täglich um 12:30 Uhr. Die Syntax für absolute Zeitangaben ist einigermaßen komplex. Die Details sind in `man systemd.time` im Abschnitt »Calendar Events« ausführlich dokumentiert. So bewirkt `OnCalendar=Sun 2016-**-* 17:15` beispielsweise, dass ein Job an jedem Sonntag des Jahrs 2016 um 17:15 Uhr ausgeführt wird.

► `OnUnitActiveSec=zeitspanne`

startet den Job neuerlich, sobald die angegebene Zeitspanne nach dem letzten Start des Jobs vergangen ist.

► `OnUnitInactiveSec=zeitspanne`

startet den Job neuerlich, sobald die angegebene Zeitspanne nach der letzten Erledigung des Jobs vergangen ist.

## Beispiel

Die folgende Datei `dnf-automatic.timer` kümmert sich darum, dass eine Stunde nach dem Bootvorgang und in der Folge einmal täglich der in der Datei `dnf-automatic.service` beschriebene Dienst ausgeführt wird. (Beide Dateien sind Bestandteil des Fedora-Pakets `dnf-automatic`, das automatisch Updates mit `dnf` durchführt.)

```
# Datei /usr/lib/systemd/system/dnf-automatic.timer
# (Fedora 23, wenn das Paket dnf-automatic installiert ist)
[Unit]
Description=dnf-automatic timer

[Timer]
OnBootSec=1h
OnUnitInactiveSec=1d

[Install]
WantedBy=basic.target
```

## /etc/vconsole.conf

Die Datei `/etc/vconsole.conf` enthält bei Distributionen, die eine aktuelle Systemd-Version als Init-System verwenden, die Tastatur- und Font-Einstellungen für die Arbeit in Textkonsolen. Bei der Veränderung der Einstellungen hilft das Kommando `localectl`. Dieses Kommando hat gegenüber einer direkten Veränderung von `vconsole.conf` den Vorteil, dass es das gewünschte Tastaturlayout auch in `/etc/X11/xorg.conf/00-keyboard.conf` speichert, sodass die Einstellung auch für das Grafiksystem gilt.

`vconsole.conf` besteht üblicherweise nur aus zwei Zeilen mit den folgenden Parametern:

► **FONT=name**

gibt die Schriftart an, die in Textkonsolen verwendet werden soll. Passende Font-Dateien befinden sich bei den meisten Distributionen im Verzeichnis `/lib/kbd/consolefonts`.

► **KEYMAP=name**

gibt das Tastaturlayout an. Die zur Auswahl stehenden Einstellungen können mit `localectl list-keymaps` ermittelt werden.

Je nach Distribution wird das Tastaturlayout für Textkonsolen an anderen Orten gespeichert:

Debian 7: /etc/default/console-setup

RHEL 6: /etc/sysconfig/keyboard

## Beispiel

Bei einer deutschen Fedora-Installation enthält `vconsole.conf` diese Einstellungen:

```
KEYMAP="de-nodeadkeys"
```

```
FONT="eurlatgr"
```

### `/etc/X11/xorg.conf`

Die Dateien `/etc/X11/xorg.conf` und `/etc/X11/xorg.conf.d/*.conf` steuern zusammen die Konfiguration des Grafiksystems X. Bis vor wenigen Jahren war eine korrekte `xorg.conf`-Datei eine Grundvoraussetzung für den Start des Grafiksystems. Mittlerweile ist `xorg.conf` hingegen optional: Der X-Server ermittelt beim Start die Parameter aller Hardware-Komponenten und führt die Konfiguration selbstständig durch. Bei Bedarf können viele Einstellungen im laufenden Betrieb durch das Kommando `xrandr` verändert werden. Das gilt insbesondere für die Bildschirmauflösung sowie für die Konfiguration mehrerer Bildschirme.

Nur wenn all das nicht funktioniert, z. B. in virtuellen Maschinen oder bei exotischen Hardware-Komponenten, ist eine manuelle Konfiguration erforderlich. Diese müssen Sie in einer Textkonsole durchführen, weil das Grafiksystem ja noch nicht funktioniert. Insofern erschien es mir zweckmäßig, die Syntax von `xorg.conf` hier zumindest kurz zusammenzufassen, auch wenn sich das Buch in erster Linie an Linux-Anwender richtet, die im Terminal arbeiten.

Abschnitt	Bedeutung
Monitor	Monitordaten (Auflösung, Bildfrequenz)
Device	Konfiguration der Grafikkarte
Screen	Bildschirmauflösung, Multi-Screen-Setup
Files	Dateinamen (z. B. Font-Verzeichnisse)
Module	Zusatzmodule (z. B. freetype, dri)
ServerFlags	verschiedene Server-Optionen
InputClass	Device-Gruppen (z. B. alle Tastaturen)
InputDevice	einzelne Eingabegeräte (z. B. Tastatur, Maus, Touchpad)

Tabelle 5 »xorg.conf«-Abschnitte



Die Datei `/etc/X11/xorg.conf` ist in mehrere Abschnitte gegliedert, die mit Section "name" eingeleitet und mit EndSection abgeschlossen werden (siehe [Tabelle 5](#)). In jedem dieser Abschnitte können einzelne Parameter in der Form Schlüsselwort "Einstellung" angegeben werden. Die Identifier-Zeile gibt dem Abschnitt einen Namen und ermöglicht Querverweise zwischen den Abschnitten. Beispielsweise verweist der Abschnitt Screen auf zumindest eine Grafikkarte (Device) und einen Monitor, im folgenden Beispiel auf `mon0` und `dev0`. Eine Referenz der unzähligen weiteren Parameter gibt man `xorg.conf`.

### Beispiel

Die folgenden Zeilen stellen eine Minimalkonfiguration für Notfälle dar, wenn das Grafiksystem gar nicht funktionieren will. Die hier vorgeschlagene Konfiguration verwendet den VESA-Treiber und bietet damit keine 3D-Unterstützung.

```
Section "Monitor"
    Identifier "mon0"
    HorizSync 31 - 94
    VertRefresh 60
EndSection
Section "Device"
    Identifier "dev0"
    Driver "vesa"
EndSection
Section "Screen"
    Identifier "screen0"
    Monitor "mon0"
    Device "dev0"
    DefaultDepth 24
    SubSection "Display"
        Depth 24
        Modes "1024x768"
    EndSubSection
EndSection
```

### `/etc/yum.conf`

Die Grundeinstellungen des unter Fedora und Red Hat üblichen Paketverwaltungssystems YUM befinden sich in `/etc/yum.conf`. Außerdem enthalten die Dateien `/etc/yum.repos.d/*.repos` Detailinstellungen für die einzelnen Paketquellen. Einige der im Folgenden beschriebenen Schlüsselwörter gelten auch für `/etc/dnf/dnf.conf`.

### Grundeinstellungen

- ▶ `clean_requirements_on_remove = true/false`

gibt an, ob Pakete, die aufgrund von Abhängigkeiten installiert wurden, automatisch wieder entfernt werden sollen, sobald alle Pakete deinstalliert werden, die auf das Paket verweisen. Auch wenn diese Option `true` lautet, entfernt `yum` nie ungefragt Pakete, die explizit mit `yum install` installiert wurden.
- ▶ `color=never`

bewirkt, dass das Kommando `yum` im Terminal keine Farben nutzt. Wenn Sie das möchten, verwenden Sie `color=always`.
- ▶ `exactarch=1`

bewirkt, dass Yum nur Updates berücksichtigt, bei denen die Architektur mit dem bereits installierten Paket übereinstimmt. i386-Pakete können also nicht durch neuere x86\_64-Pakete ersetzt werden.
- ▶ `gpgcheck=1`

bewirkt, dass Yum mit einem Schlüssel die Authentizität der Pakete sicherstellt. `gpgcheck` kann abweichend von der Einstellung in `yum.conf` auch individuell für jede Paketquelle eingestellt werden. `plugins` entscheidet, ob Yum Plug-ins berücksichtigt.
- ▶ `installonly_limit=n`

steuert, wie viele alte Versionen von Kernelpaketen gleichzeitig installiert werden. Die Einstellung betrifft alle Pakete, zu denen Updates in Form von neuen Paketen installiert werden. Diese Vorgehensweise stellt sicher, dass immer auch ältere Versionen des betroffenen Pakets als Fallback zur Verfügung stehen.
- ▶ `keepcache=0`

bewirkt, dass heruntergeladene Pakete nach der Installation nicht archiviert werden. In der Regel ist das eine zweckmäßige Einstellung, weil der Platzbedarf für die Pakete im Laufe der Zeit recht groß wird und normalerweise kein Grund dazu besteht, die Pakete ein zweites Mal zu installieren. Es kann allerdings passieren, dass `yum` während der Installation ein Problem feststellt und die Installation abbricht. Auch in diesem Fall werden die heruntergeladenen Pakete gelöscht. Wenn Sie das Problem beheben können und anschließend das Update wiederholen, müssen alle Pakete neuerlich heruntergeladen werden. Diese Situation vermeiden Sie mit `keepcache=1`. Um die heruntergeladenen Pakete in `/var/cache/yum` explizit zu löschen, führen Sie `yum clean packages` aus.

## Paketquellen

Die Konfiguration der Paketquellen in `/etc/repos.d/*.repo` gilt sowohl für `yum` als auch für das modernere Paketverwaltungskommando `dnf`, das unter Fedora zum Einsatz kommt.

- ▶ `[repovariante]`  
gibt an, auf welchen Teilbereich der Paketquelle sich die folgenden Einstellungen beziehen.
- ▶ `baseurl=url` oder `mirrorlist=url`  
gibt an, wo sich die Paketdateien befinden. Bei `mirrorlist` verweist die `url` auf eine Liste von Mirror-Servern. Yum entscheidet sich selbstständig für einen der Mirrors. Yum ersetzt in der `url` die Variablen `$releasever`, `$arch` und `$basearch` durch die Versionsnummer der Linux-Distribution und deren Architektur.
- ▶ `enabled=0/1`  
gibt an, ob die Paketquelle aktiv ist.
- ▶ `name=reponame`  
gibt den Namen der Paketquelle an.
- ▶ `metadata_expires`  
steuert, wie lange die von einer Paketquelle heruntergeladenen Metadaten gültig sind. Yum speichert die Metadaten in einem Cache und verzichtet auf ein neuerliches Herunterladen, wenn die Metadaten noch nicht veraltet sind. Das spart Zeit und Download-Volumen, kann aber dazu führen, dass Yum kürzlich durchgeführte Änderungen in der Paketquelle ignoriert. Gegebenenfalls erzwingen Sie durch `yum clean metadata` das Löschen der lokalen Metadaten. Damit ist Yum beim nächsten Mal gezwungen, die Metadaten aller Paketquellen neu einzulesen.

## Beispiel

Die folgenden Zeilen zeigen einen Ausschnitt aus der Datei `/etc/yum.repos.d/fedora.repo`. Diese Datei enthält die Definition der Paketquelle mit den Basispaketen der Fedora-Distribution:

```
[fedora]
name=Fedora $releasever - $basearch
failovermethod=priority
mirrorlist=https://mirrors.fedoraproject.org/\
    metalink?repo=fedora-$releasever&arch=$basearch
enabled=1
```

```
metadata_expire=7d
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora-$releasever-$basearch
```

```
[fedora-source]
name=Fedora $releasever - Source
mirrorlist=https://mirrors.fedoraproject.org/\
    metalink?repo=fedora-source-$releasever&arch=$basearch
enabled=0
...
```




# Tastenkürzel

Im letzten Abschnitt der *Linux-Kommandoreferenz* geht es um die Tastenkürzel der wichtigsten Editoren und anderer Kommandos, die üblicherweise über die Tastatur bedient werden. Dazu zählen z. B. `bash`, `man`, `info`, `less` und `mutt`.

Nahezu alle Programme bieten die Möglichkeit, eigene Tastenkürzel zu definieren. Dieser Abschnitt bezieht sich auf die Defaultkonfiguration, die bei den meisten Linux-Distributionen standardmäßig gilt.

## bash

Tabelle 6 fasst zusammen, welche Tastenkürzel Sie innerhalb der Bourne Again Shell (`bash`) bei der Eingabe von Kommandos verwenden können. Die Tastenkürzel stammen eigentlich von der `readline`-Bibliothek. Die Konfiguration dieser Bibliothek können Sie in `/etc/inputrc` bzw. `.inputrc` verändern.

Tastenkürzel	Funktion
<code>Strg</code> + <code>A</code>	bewegt den Cursor an den Zeilenanfang (wie <code>Pos1</code> ).
<code>Strg</code> + <code>C</code>	bricht das laufende Kommando ab.
<code>Strg</code> + <code>E</code>	setzt den Cursor an das Ende der Zeile (wie <code>Ende</code> ).
<code>Strg</code> + <code>K</code>	löscht den Rest der Zeile ab der Cursorposition.
<code>Strg</code> + <code>Y</code>	fügt den zuletzt gelöschten Text wieder ein.
<code>Strg</code> + <code>Z</code>	unterbricht das laufende Kommando (Fortsetzung mit <code>fg</code> oder <code>bg</code> ).
	vervollständigt Datei- und Kommandonamen.
 / 	blättert durch die bisher ausgeführten Kommandos.

**Tabelle 6** Tastenkürzel zur Kommandoeingabe in der `bash`

## emacs

Der Emacs zählt zu den funktionsreichsten und komplexesten Editoren, die unter Linux zur Verfügung stehen. Es gibt Hunderte von Tastenkürzeln und Kommandos, von denen hier natürlich nur die allerwichtigsten präsentiert werden.

Generell gibt es drei Möglichkeiten zur Eingabe von Emacs-Kommandos: das Menü, die Verwendung von Tastenkürzeln (zumeist eine Kombination mit `Strg`) oder `Alt`) oder die Eingabe des gesamten Kommandonamens. Die dritte Variante wird mit `Alt+X` oder `Esc` eingeleitet, also etwa `Alt+X` delete-char `↵`. Die Eingabe von Kommandos und anderen Parametern wird durch zwei Mechanismen erleichtert:

- ▶ Während der Eingabe können Sie den Namen eines Emacs-Kommandos mit `ESC` ergänzen. In gleicher Weise können auch Dateinamen ergänzt werden.
- ▶ Auf früher bei `Alt+X` angegebene Kommandos können Sie nach der Einleitung des neuen Kommandos durch `Alt+X` mit `Alt+P` (Previous) und `Alt+N` (Next) zurückgreifen.

In der Dokumentation zum Emacs werden Tastenkürzel etwas abweichend dargestellt: DEL bedeutet nicht `Entf`, sondern `↵`! C steht für Control (gemeint ist `Strg`) und M für `Meta`. Eine direkte Entsprechung der Meta-Taste existiert auf einer Standard-PC-Tastatur nicht. M-x kann auf einer PC-Tastatur auf zwei Weisen nachgebildet werden: durch `Esc` und `X` (nacheinander) oder durch `Alt+X`.

Tabelle 7 fasst die Kommandos für die Grundfunktionen zusammen. Zur Cursorbewegung können Sie außer den Cursortasten auch diverse Tastenkürzel verwenden, die in Tabelle 8 angegeben sind. Tabelle 9 gibt an, wie Sie Text markieren, löschen und wieder einfügen, und Tabelle 10 fasst zusammen, wie Sie suchen und ersetzen.

Tastenkürzel	Funktion
<code>Strg+X</code> , <code>Strg+F</code>	lädt eine neue Datei.
<code>Strg+X</code> , <code>Strg+S</code>	speichert die aktuelle Datei.
<code>Strg+X</code> , <code>Strg+W</code>	speichert die Datei unter einem neuen Namen.
<code>Strg+G</code>	bricht die Eingabe eines Kommandos ab.
<code>Strg+X</code> , <code>U</code>	macht die letzte Änderung rückgängig (Undo).
<code>Strg+X</code> , <code>Strg+C</code>	beendet den Emacs (mit Rückfrage zum Speichern).

Tabelle 7 Elementare Emacs-Kommandos

Tastenkürzel	Funktion
<code>Alt+F</code> / <code>Alt+B</code>	bewegt den Cursor ein Wort vor bzw. zurück.
<code>Strg+A</code> / <code>Strg+E</code>	stellt den Cursor an den Beginn bzw. das Ende der Zeile.
<code>Alt+&lt;</code> / <code>Alt+⇧+&gt;</code>	bewegt den Cursor an den Beginn bzw. das Ende des Textes.
<code>Alt+G</code> <i>n</i> <code>↵</code>	bewegt den Cursor in Zeile <i>n</i> .

Tabelle 8 Cursorbewegung

Tastenkürzel	Funktion
<code>Strg</code> +Leertaste	setzt einen (unsichtbaren) Markierungspunkt.
<code>Strg</code> + <code>W</code>	löscht den Text zwischen dem Markierungspunkt und der aktuellen Cursorposition.
<code>Strg</code> + <code>Y</code>	fügt den gelöschten Text wieder ein.
<code>Strg</code> + <code>X</code> , <code>Strg</code> + <code>X</code>	vertauscht Cursorposition und Markierungspunkt.
<code>Alt</code> + <code>D</code>	löscht das nächste Wort bzw. das Ende des Wortes ab dem Cursor.
<code>Alt</code> + <code>←</code>	löscht das vorige Wort bzw. den Beginn des Wortes bis zum Cursor.
<code>Strg</code> + <code>K</code>	löscht den Rest der Zeile ab der Cursorposition.
<code>Alt</code> + <code>0</code> , <code>Strg</code> + <code>K</code>	löscht den Zeilenanfang vor der Cursorposition.
<code>Alt</code> + <code>M</code>	löscht den nächsten Absatz.
<code>Alt</code> + <code>Z</code> , <code>x</code>	löscht alle Zeichen bis zum nächsten Auftreten von <code>x</code> . Das Zeichen <code>x</code> wird mitgelöscht.
<code>Strg</code> + <code>Y</code>	fügt den zuletzt gelöschten Text an der Cursorposition wieder ein.

**Tabelle 9** Text markieren, löschen und wieder einfügen

Tastenkürzel	Funktion
<code>Strg</code> + <code>S</code>	inkrementelle Suche vorwärts
<code>Strg</code> + <code>R</code>	inkrementelle Suche rückwärts
<code>Alt</code> + <code>P</code>	wählt einen früher verwendeten Suchtext aus (Previous).
<code>Alt</code> + <code>N</code>	wählt einen später verwendeten Suchtext aus (Next).
<code>Strg</code> + <code>G</code>	Abbruch der Suche
<code>Strg</code> + <code>X</code> , <code>Strg</code> + <code>X</code>	vertauscht den Markierungspunkt (Beginn der Suche) und die aktuelle Cursorposition.
<code>Strg</code> + <code>Alt</code> + <code>S</code>	inkrementelle Mustersuche vorwärts
<code>Strg</code> + <code>Alt</code> + <code>R</code>	inkrementelle Mustersuche rückwärts
<code>Alt</code> + <code>%</code>	Suchen und Ersetzen ohne Muster
<code>Alt</code> + <code>X</code> query-replace-r <code>↵</code>	Suchen und Ersetzen mit Muster

**Tabelle 10** Suchen und Ersetzen

**fdisk**

fdisk ist ein interaktives Programm zur Partitionierung von Festplatten. Eine Beschreibung der wichtigsten Optionen sowie ein längeres Anwendungsbeispiel finden Sie in der alphabetischen Kommandoreferenz. Tabelle 11 fasst lediglich die Tastenkürzel zur Bedienung des Programms zusammen.

Tastenkürzel	Bedeutung
[D]	Partition löschen ( <i>delete</i> )
[L]	Partitions-ID-Nummer anzeigen ( <i>list</i> )
[M]	Online-Hilfe ( <i>menu</i> )
[N]	neue Partition anlegen ( <i>new</i> )
[P]	Partitionsliste anzeigen ( <i>print</i> )
[Q]	Programm beenden (ohne die Partitionstabelle zu verändern; <i>quit</i> )
[T]	Partitionstyp verändern
[U]	Maßeinheit zwischen Zylindern und Sektoren umschalten ( <i>unit</i> )
[V]	Partitionstabelle überprüfen ( <i>verify</i> )
[W]	Partitionstabelle ändern ( <i>write</i> )

Tabelle 11 fdisk-Tastenkürzel

**gnome-terminal**

Wenn Sie Shell-Kommandos unter Gnome ausführen, verwenden Sie dazu höchstwahrscheinlich das Programm `gnome-terminal`. Damit Sie die in der `bash` üblichen Tastenkürzel verwenden können, sollten Sie als Erstes mit BEARBEITEN • EINSTELLUNGEN die Option MENÜKÜRZELBUCHSTABEN zur Steuerung der Menüs durch [Alt]-Tastenkürzel deaktivieren.

Zur Menüsteuerung können Sie dann bei Bedarf immer noch die Taste [F10] verwenden – es sei denn, Sie deaktivieren auch die Verarbeitung dieser Taste im gerade erwähnten Konfigurationsdialog. Einige `gnome-terminal`-spezifische Tastenkürzel bleiben auf jeden Fall verfügbar; sie sind in Tabelle 12 zusammengefasst.
















Tastenkürzel	Funktion
 + <b>Strg</b> + <b>C</b>	kopiert den markierten Text in die Zwischenablage.
 + <b>Strg</b> + <b>F</b>	sucht einen Text in den Terminalausgaben.
 + <b>Strg</b> + <b>G</b>	wiederholt die Suche rückwärts.
 + <b>Strg</b> + <b>H</b>	wiederholt die Suche vorwärts.
 + <b>Strg</b> + <b>N</b>	öffnet ein neues Terminalfenster.
 + <b>Strg</b> + <b>Q</b>	schließt das Fenster.
 + <b>Strg</b> + <b>T</b>	öffnet einen neuen Terminal-Reiter.
 + <b>Strg</b> + <b>V</b>	fügt den Inhalt der Zwischenablage ein.
 + <b>Strg</b> + <b>W</b>	schließt den Reiter.
 + <b>Strg</b> + <b>+</b>	vergrößert die Schrift.
 + <b>Strg</b> + <b>-</b>	verkleinert die Schrift.
 + <b>Strg</b> + <b>Bild↑</b> / <b>Bild↓</b>	wechselt in den vorigen/nächsten Reiter.
<b>F11</b>	aktiviert bzw. deaktiviert den Vollbildmodus.

Tabelle 12 Tastenkürzel in »gnome-terminal«

## grub

Im Linux-Bootloader GRUB können Sie mit den Cursortasten ein Betriebssystem bzw. eine Linux-Variante auswählen und diese dann durch  starten. Darüber hinaus bietet GRUB die Möglichkeit, die Parameter eines Menüeintrags interaktiv zu verändern oder eigene Kommandos auszuführen. [Tabelle 13](#) fasst hierfür die wichtigsten Tastenkürzel zusammen. Die Tabelle bezieht sich dabei auf die aktuelle GRUB-Version 2.


Tastenkürzel	Funktion
<b>Esc</b>	beendet den Grafikmodus und aktiviert den Textmodus.
<b>C</b>	startet den Kommandomodus zur interaktiven Ausführung von GRUB-Kommandos. Bei der Kommandoeingabe können Dateinamen wie in der Shell durch  vervollständigt werden.
<b>E</b>	startet den Editor für den ausgewählten Menüeintrag.

Tabelle 13 Tastenkürzel zur interaktiven Steuerung von GRUB 2

Tastenkürzel	Funktion
<b>P</b>	gibt die interaktiven GRUB-Funktionen durch die Eingabe eines Passworts frei. Das ist nur erforderlich, wenn GRUB durch ein Passwort abgesichert ist.
<b>Strg</b> + <b>X</b> oder <b>F10</b>	startet den zuvor mit <b>E</b> veränderten Menüeintrag.

Tabelle 13 Tastenkürzel zur interaktiven Steuerung von GRUB 2 (Forts.)

info

`info` startet das gleichnamige Online-Hilfesystem. Zur Navigation im Hilfetext verwenden Sie die in [Tabelle 14](#) zusammengefassten Tastenkürzel. `info`-Texte können Sie mit mehr Komfort auch mit dem Kommando `pinfo` aus dem gleichnamigen Paket, mit dem Editor Emacs oder in den Hilfesystemen von Gnome und KDE lesen.




Tastenkürzel	Funktion
Leertaste	scrollt Text nach unten.
	scrollt Text nach oben.
<b>B</b> , <b>E</b>	springt zum Anfang/Ende der Info-Einheit ( <i>beginning/end</i> ).
	bewegt den Cursor zum nächsten Querverweis.
	verfolgt einen Querverweis zu einer anderen Info-Einheit.
<b>N</b>	zeigt die nächste Info-Einheit in derselben Hierarchiestufe an ( <i>next</i> ).
<b>P</b>	zeigt die vorige Info-Einheit in derselben Hierarchiestufe an ( <i>previous</i> ).
<b>U</b>	springt eine Hierarchieebene nach oben ( <i>up</i> ).
<b>L</b>	springt zurück zum zuletzt angezeigten Text ( <i>last</i> ).
<b>H</b>	zeigt eine ausführliche Bedienungsanleitung an ( <i>help</i> ).
<b>?</b>	zeigt eine Kommandoübersicht an.
<b>Q</b>	beendet <code>info</code> ( <i>quit</i> ).

Tabelle 14 »info«-Tastenkürzel

## joe

joe ist ein einfacher Editor, dessen Tastenkürzel dem Textverarbeitungsprogramm Wordstar nachempfunden sind (siehe [Tabelle 15](#)). Der Editor kann auch unter den Namen jmacs oder jpico gestartet werden. Es gelten dann andere Tastenkürzel, die zum Emacs bzw. zu Pico kompatibel sind.

Tastenkürzel	Funktion
<b>Strg</b> + <b>K</b> , <b>H</b>	blendet das Hilfefenster ein/aus.
<b>Strg</b> + <b>K</b> , <b>E</b>	lädt eine neue Datei.
<b>Strg</b> + <b>K</b> , <b>D</b>	speichert die Datei (wahlweise unter neuem Namen).
<b>Strg</b> + <b>Y</b>	löscht eine Zeile.
<b>Strg</b> + <b>⇧</b> , <b>-</b>	macht das Löschen rückgängig (Undo).
<b>Strg</b> + <b>C</b>	beendet joe (mit Rückfrage zum Speichern).

**Tabelle 15** »joe«-Tastenkürzel

## konsole

KDE-Fans führen Shell-Kommandos in der Regel im Programm konsole aus. Die meisten Tastenkürzel werden von diesem Programm direkt an die Shell weitergegeben. Darüber hinaus gibt es aber einige konsole-spezifische Tastenkürzel, die in [Tabelle 16](#) zusammengefasst sind.

Tastenkürzel	Funktion
<b>⇧</b> + <b>Strg</b> + <b>A</b>	markiert den Reiter bei Aktivität.
<b>⇧</b> + <b>Strg</b> + <b>C</b>	kopiert den markierten Text in die Zwischenablage.
<b>⇧</b> + <b>Strg</b> + <b>F</b>	sucht einen Text in den Terminalausgaben.
<b>F3</b>	wiederholt die Suche rückwärts.
<b>⇧</b> + <b>F3</b>	wiederholt die Suche vorwärts.
<b>⇧</b> + <b>Strg</b> + <b>I</b>	markiert den Reiter bei längerer Inaktivität.
<b>⇧</b> + <b>Strg</b> + <b>N</b>	öffnet ein neues Terminalfenster.
<b>⇧</b> + <b>Strg</b> + <b>Q</b>	schließt das Fenster.

**Tabelle 16** Tastenkürzel in »konsole«







Tastenkürzel	Funktion
 + <b>Strg</b> + <b>T</b>	öffnet einen neuen Terminal-Reiter.
 + <b>Strg</b> + <b>V</b>	fügt den Inhalt der Zwischenablage ein.
 + <b>Strg</b> + <b>W</b>	schließt den Reiter.
 + <b>Strg</b> + <b>+</b>	vergrößert die Schrift.
 + <b>Strg</b> + <b>-</b>	verkleinert die Schrift.
 + <b>Strg</b> + <b>←</b> / <b>→</b>	wechselt in den vorigen/nächsten Reiter.

Tabelle 16 Tastenkürzel in »konsole« (Forts.)

## less

Das Kommando `less` zeigt Texte an. Während das Programm läuft, können Sie mit den Cursortasten durch den Text scrollen, Texte suchen, den durch die Umgebungsvariable `$EDITOR` eingestellten Editor starten etc. (siehe [Tabelle 17](#)). `less` wird auch zur Anzeige von `man`-Hilfetexten verwendet.





Tastenkürzel	Funktion
Cursortasten	scrollt den Text nach oben oder unten.
<b>Pos1</b> , <b>Ende</b>	springt an den Beginn bzw. das Ende des Textes.
<b>G</b> ,  + <b>G</b>	springt an den Beginn bzw. das Ende des Textes.
<b>/</b> muster  <b>↵</b>	sucht vorwärts.
<b>?</b> muster  <b>↵</b>	sucht rückwärts.
<b>N</b>	wiederholt die Suche vorwärts ( <i>next</i> ).
 + <b>N</b>	wiederholt die Suche rückwärts.
<b>V</b>	startet den durch <code>\$EDITOR</code> oder <code>\$VISUAL</code> eingestellten Editor.
<b>Q</b>	beendet <code>less</code> ( <i>quit</i> ).
<b>H</b>	zeigt einen Hilfetext mit weiteren Tastenkürzeln an.








Tabelle 17 »less«-Tastenkürzel

**man**

Das Kommando man zeigt die Dokumentation zu wichtigen Kommandos, Konfigurationsdateien, C-Funktionen etc. an. Um ein einfaches Blättern durch den Hilfetext zu ermöglichen, greift man auf das Kommando less zurück. Deswegen gelten innerhalb von man dieselben Tastenkürzel wie bei less (siehe Tabelle 17).

**mutt**

mutt ist ein E-Mail-Client für den Textmodus. Das Programm ist gut dazu geeignet, lokal auf dem Rechner gespeicherte E-Mails zu lesen und zu beantworten. Hard-Core-Linux-Fans bevorzugen mutt sogar in grafischen Benutzeroberflächen, weil sich das Programm besonders effizient per Tastatur steuern lässt. Tabelle 18 enthält nur die wichtigsten Tastenkürzel.

Tastenkürzel	Funktion
Cursortasten	bewegt den Cursor durch die Mail-Liste.
	zeigt die ersten Zeilen der ausgewählten E-Mail an.
Leertaste	blättert durch den Text einer Mail.
	löscht die E-Mail.
	wechselt von der E-Mail-Ansicht zurück in die Inbox.
	zeigt die nächste E-Mail an.
	verfasst eine neue E-Mail mit dem durch \$EDITOR oder \$VISUAL eingestellten Editor.
	beantwortet die E-Mail.
	zeigt einen Hilfetext mit allen Tastenkürzeln an.

**Tabelle 18** »mutt«-Tastenkürzel

**nano**

nano ist ein minimalistischer Editor, der vor allem für Einsteiger gut geeignet ist. Die wichtigsten Tastenkürzel (siehe auch Tabelle 19) werden ständig in den zwei untersten Textzeilen eingeblendet.

Tastenkürzel	Funktion
<b>Strg</b> + <b>A</b>	bewegt den Cursor an den Beginn der Zeile.
<b>Strg</b> + <b>D</b>	löscht ein Zeichen.
<b>Strg</b> + <b>E</b>	bewegt den Cursor zum Ende der Zeile.
<b>Strg</b> + <b>H</b>	löscht ein Zeichen rückwärts.
<b>Strg</b> + <b>^</b>	setzt einen Markierungspunkt.
<b>Strg</b> + <b>K</b>	löscht die aktuelle Zeile oder den markierten Text.
<b>Strg</b> + <b>U</b>	fügt den gelöschten Text wieder ein.
<b>Strg</b> + <b>R</b>	fügt eine Textdatei in den Text ein.
<b>Strg</b> + <b>O</b>	speichert die Datei.
<b>Strg</b> + <b>X</b>	beendet den Editor.

Tabelle 19 »nano«-Tastenkürzel

## Textkonsole

Wenn Sie direkt in Textkonsolen arbeiten, also nicht unter KDE, Gnome oder einem anderen Desktop-System und auch nicht via SSH, dann gelten dort einige besondere Tastenkürzel. Tabelle 20 fasst die wichtigsten Kürzel zusammen.

Tastenkürzel	Funktion
<b>Strg</b> + <b>Alt</b> + <b>Fn</b>	wechselt vom Grafikmodus in die Textkonsole <i>n</i> .
<b>Alt</b> + <b>Fn</b>	wechselt von einer Textkonsole in eine andere Textkonsole <i>n</i> .
<b>Alt</b> + <b>F7</b>	wechselt zurück in den Grafikmodus ( <b>Alt</b> + <b>F1</b> bei Fedora, <b>Alt</b> + <b>F5</b> bei Knoppix).
<b>Alt</b> + <b>→</b> / <b>←</b>	wechselt in die vorige/nächste Textkonsole.
<b>⇧</b> + <b>Bild↑</b> / <b>Bild↓</b>	scrollt seitenweise vorwärts/rückwärts.
<b>Strg</b> + <b>Alt</b> + <b>Entf</b>	beendet Linux durch shutdown (Vorsicht!).

Tabelle 20 Tastenkürzel in Textkonsolen

## vi/vim

Der Editor Vi ist ein Urgestein der Unix-Geschichte. Nahezu alle Linux-Distributionen installieren standardmäßig das zum Vi kompatible Programm `vim`, das Sie gleichermaßen mit `vi` oder mit `vim` starten können.

Der Vi bietet ähnlich viele Funktionen wie der Emacs, die Bedienung ist aber noch schwieriger zu erlernen. [Tabelle 21](#) fasst die elementarsten Vi-Kommandos zusammen.

Tastenkürzel	Funktion
<code>I</code>	aktiviert den Einfügemodus.
<code>A</code>	aktiviert den Einfügemodus. Die Texteingabe beginnt beim nächsten Zeichen.
<code>Esc</code>	aktiviert den Standardmodus bzw. bricht die Kommandoeingabe ab.
<b>Kommandos im Standardmodus</b>	
<code>D</code> , <code>W</code>	löscht ein Wort.
<code>D</code> , <code>D</code>	löscht die aktuelle Zeile.
<code>n D</code> , <code>D</code>	löscht <i>n</i> Zeilen.
<code>P</code>	fügt den zuletzt gelöschten Text hinter der Cursorposition ein.
<code>⇧+P</code>	fügt den zuletzt gelöschten Text vor der Cursorposition ein.
<code>.</code>	wiederholt das letzte Kommando.
<code>U</code>	macht die letzte Änderung rückgängig (Undo).
<code>⇧+U</code>	widerruft alle Änderungen in der aktuellen Zeile.
<code>Strg+R</code>	macht Undo rückgängig (Redo, ab Vim 7).
<code>:w</code>	speichert die Datei.
<code>:q</code>	beendet <code>vim</code> .
<code>:q!</code>	beendet <code>vim</code> auch dann, wenn es nicht gespeicherte Dateien gibt
<b>Kommandos im Einfügemodus</b>	
<code>Strg+O</code> <i>kommando</i>	führt das Kommando aus, ohne den Einfügemodus zu verlassen.

**Tabelle 21** Elementare Kommandos

Der wichtigste fundamentale Unterschied zu anderen Editoren besteht darin, dass der Vi zwischen verschiedenen Modi unterscheidet:

- **Insert-Modus:** Um Text einzugeben, müssen Sie mit **I** (*insert*) oder **A** (*append*) in den Einfügemodus wechseln. `vim` zeigt nun in der untersten Zeile ganz links den Text `-- EINFÜGEN --` an. Im Einfügemodus können Sie Text eingeben, den Cursor bewegen und einzelne Zeichen löschen (**Entf** und **←**). Der Unterschied zwischen **I** und **A** besteht darin, dass die Eingabe bei **I** an der aktuellen Cursorposition beginnt, bei **A** beim Zeichen dahinter.
- **Complex-Command-Modus:** Die Eingabe der meisten Kommandos erfolgt hingegen im Complex-Command-Modus, der mit **:** aktiviert wird. Vorher muss gegebenenfalls der Insert-Modus durch **Esc** verlassen werden.

Tabelle 22 listet nützliche Kommandos zur Cursorbewegung auf. In Tabelle 23, Tabelle 24, Tabelle 25 und Tabelle 26 geht es darum, wie Sie Text löschen, kopieren, markieren und bearbeiten.

Tastenkürzel	Funktion
Cursortasten	Die Cursortasten haben die übliche Bedeutung.
<b>H</b> / <b>L</b>	bewegt den Cursor nach links/rechts.
<b>J</b> / <b>K</b>	bewegt den Cursor nach unten/oben.
<b>⇧</b> + <b>H</b> / <b>⇧</b> + <b>L</b>	bewegt den Cursor an den Beginn bzw. das Ende der aktuellen Seite.
<b>⇧</b> + <b>M</b>	bewegt den Cursor in die Mitte der aktuellen Seite.
<b>B</b> / <b>W</b>	bewegt den Cursor um ein Wort nach links/rechts.
<b>E</b>	bewegt den Cursor an das Ende des Worts.
<b>G</b> , <b>E</b>	bewegt den Cursor an den Anfang des Worts.
<b>(</b> , <b>)</b>	bewegt den Cursor an den Beginn des aktuellen/nächsten Satzes.
<b>{</b> , <b>}</b>	bewegt den Cursor an den Beginn des aktuellen/nächsten Absatzes.
<b>^</b> , <b>\$</b>	bewegt den Cursor an den Beginn bzw. das Ende der Zeile.
<b>⇧</b> + <b>G</b>	bewegt den Cursor an das Ende der Datei.
<b>G</b> , <b>G</b>	bewegt den Cursor an den Beginn der Datei.
<i>n</i> <b>⇧</b> + <b>G</b>	bewegt den Cursor in die Zeile <i>n</i> .
<i>n</i> <b> </b>	bewegt den Cursor in die Spalte <i>n</i> .
<b>%</b>	bewegt den Cursor zum korrespondierenden Klammerzeichen <b>()[]{}.</b>

**Tabelle 22** Tastenkürzel zur Cursorbewegung im Standardmodus



Tastenkürzel im Einfügemodus	
[Entf], [←]	Diese Tasten haben die übliche Bedeutung.
Kommandos im Standardmodus	
[X]	löscht das Zeichen an der Cursorposition bzw. den markierten Text
[⇧] + [X]	löscht das Zeichen vor dem Cursor.
[D], [D]	löscht die aktuelle Zeile.
[D] <i>cursorkommando</i>	löscht den Text entsprechend dem Kommando zur Cursorbewegung (siehe <a href="#">Tabelle 22</a> ). Beispiele: [D], [§] löscht bis zum Ende der Zeile. [D], [B] löscht das vorige Wort. [D], [W] löscht das nächste Wort.

Tabelle 23 Text löschen

Tastenkürzel	Funktion
[Y]	kopiert den markierten Text in das Kopierregister.
[Y], [Y]	kopiert die aktuelle Zeile in das Kopierregister.
[Y] <i>cursorkommando</i>	kopiert den durch die Cursorbewegung erfassten Text; Beispiel: [Y], [J] kopiert den Text bis zum Ende des Absatzes.

Tabelle 24 Text in das Kopierregister kopieren

Tastenkürzel	Funktion
[V]	(de)aktiviert den Zeichenmarkierungsmodus.
[⇧] + [V]	(de)aktiviert den Zeilenmarkierungsmodus.
[Strg] + [V]	(de)aktiviert den Blockmarkierungsmodus.
[A], [W]	vergrößert die Markierung um ein Wort.
[A], [S]	vergrößert die Markierung um einen Satz.
[A], [P]	vergrößert die Markierung um einen Absatz.
[A], [B]	vergrößert die Markierung um eine ()-Ebene.
[A], [⇧] + [B]	vergrößert die Markierung um eine {}-Ebene.
[G], [V]	markiert den zuletzt markierten Text nochmals.
[O]	wechselt die Cursorposition zwischen Markierungsanfang und -ende.

Tabelle 25 Text markieren

Tastenkürzel	Funktion
[X]	löscht den markierten Text.
[Y]	kopiert den markierten Text in das Kopierregister.
[~]	ändert die Groß-/Kleinschreibung.
[J]	fügt die markierten Zeilen zu einer langen Zeile zusammen.
[G], [Q]	führt einen Zeilenumbruch durch (für Fließtext).
[>], [<]	rückt den Text um eine Tabulatorposition ein oder aus.
[=]	rückt den Text dem aktuellen indent-Modus entsprechend neu ein.
!sort	sortiert die Zeilen mit dem externen Kommando sort.

Tabelle 26 Markierten Text bearbeiten

### Suchen und ersetzen

Im Standardmodus bewegt [/ *suchtext* [↵] den Cursor zum gesuchten Text. [N] wiederholt die Suche, [↵]+[N] wiederholt die Suche rückwärts. Um von vornherein rückwärts zu suchen, beginnen Sie die Suche mit [?] *suchausdruck*. Tabelle 27 fasst die wichtigsten Sonderzeichen zusammen, um nach Mustern zu suchen.

Zeichen	Bedeutung
.	ein beliebiges Zeichen
^ \$	Zeilenanfang/Zeilenende
\< \>	Wortanfang/Wortende
[a-e]	ein Zeichen zwischen <i>a</i> und <i>e</i>
\s, \t	ein Leerzeichen bzw. ein Tabulatorzeichen
\( \)	fasst ein Suchmuster als Gruppe zusammen.
\=	Der Suchausdruck muss 0- oder einmal auftreten.
*	Der Suchausdruck darf beliebig oft (auch 0-mal) auftreten.
\+	Der Suchausdruck muss mindestens einmal auftreten.

Tabelle 27 Sonderzeichen im Suchausdruck

Vim unterscheidet bei der Suche zwischen Groß- und Kleinschreibung. Wenn Sie das nicht möchten, leiten Sie das Suchmuster mit /c ein (gilt nur für diese Suche) oder führen [: set ignorecase aus (gilt für alle weiteren Suchen).

Mit `:` set `incsearch` aktivieren Sie die inkrementelle Suche: Bereits während der Eingabe des Suchtexts durch `/` *suchausdruck* bewegt Vim den Cursor zum ersten passenden Ort. `\` beendet die Suche, `Esc` bricht sie ab. Nach der Suche bleiben alle Übereinstimmungen im Text markiert, bis Sie eine neue Suche durchführen oder `:` `nohlsearch` ausführen.

Um alle Vorkommen des Texts *abc* ohne Rückfrage durch *efg* zu ersetzen, führen Sie im Standardmodus `:` `%s/abc/efg/g` aus. `'` `'` führt anschließend zurück an den Beginn der Suche. Tabelle 28 fasst einige Varianten des Suchen-und-ersetzen-Kommandos zusammen.

Tastenkürzel	Funktion
<code>:</code> <code>%s/abc/efg/g</code>	ersetzt ohne Rückfrage alle Vorkommen von <i>abc</i> durch <i>efg</i> .
<code>:</code> <code>%s/abc/efg/gc</code>	ersetzt mit Rückfrage alle Vorkommen von <i>abc</i> durch <i>efg</i> .
<code>:</code> <code>%s/abc/efg/gci</code>	ersetzt ohne Berücksichtigung der Groß- und Kleinschreibung.

Tabelle 28 Suchen und ersetzen

## Die Serviceseiten

Im Folgenden finden Sie Hinweise, wie Sie Kontakt zu uns aufnehmen können.

### Lob und Tadel

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Wenn Sie zufrieden waren, empfehlen Sie das Buch bitte weiter. Wenn Sie meinen, es gebe doch etwas zu verbessern, schreiben Sie direkt an den Lektor dieses Buches: [sebastian.kestel@rheinwerk-verlag.de](mailto:sebastian.kestel@rheinwerk-verlag.de). Wir freuen uns über jeden Verbesserungsvorschlag, aber über ein Lob freuen wir uns natürlich auch!

Auch auf unserer Webkatalogseite zu diesem Buch haben Sie die Möglichkeit, Ihr Feedback an uns zu senden oder Ihre Leseerfahrung per Facebook, Twitter oder E-Mail mit anderen zu teilen. Folgen Sie einfach diesem Link: <http://www.rheinwerk-verlag.de/3856>.

### Zusatzmaterialien

Zusatzmaterialien (Beispielcode, Übungsmaterial, Listen usw.) finden Sie in Ihrer Online-Bibliothek sowie auf der Webkatalogseite zu diesem Buch: <http://www.rheinwerk-verlag.de/3856>. Wenn uns sinnentstellende Tippfehler oder inhaltliche Mängel bekannt werden, stellen wir Ihnen dort auch eine Liste mit Korrekturen zur Verfügung.

### Technische Probleme

Im Falle von technischen Schwierigkeiten mit dem E-Book oder Ihrem E-Book-Konto beim Rheinwerk Verlag steht Ihnen gerne unser Leserservice zur Verfügung: [ebooks@rheinwerk-verlag.de](mailto:ebooks@rheinwerk-verlag.de).

## Über uns und unser Programm

Informationen zu unserem Verlag und weitere Kontaktmöglichkeiten bieten wir Ihnen auf unserer Verlagswebsite <http://www.rheinwerk-verlag.de>. Dort können Sie sich auch umfassend und aus erster Hand über unser aktuelles Verlagsprogramm informieren und alle unsere Bücher und E-Books schnell und komfortabel bestellen. Alle Buchbestellungen sind für Sie versandkostenfrei.

## Rechtliche Hinweise

In diesem Abschnitt finden Sie die ausführlichen und rechtlich verbindlichen Nutzungsbedingungen für dieses E-Book.

### Copyright-Vermerk

Das vorliegende Werk ist in all seinen Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und beim Rheinwerk Verlag. Insbesondere das Recht der Vervielfältigung und Verbreitung, sei es in gedruckter oder in elektronischer Form.

© Rheinwerk Verlag GmbH, Bonn 2016

### Ihre Rechte als Nutzer

Sie sind berechtigt, dieses E-Book ausschließlich für persönliche Zwecke zu nutzen. Insbesondere sind Sie berechtigt, das E-Book für Ihren eigenen Gebrauch auszudrucken oder eine Kopie herzustellen, sofern Sie diese Kopie auf einem von Ihnen alleine und persönlich genutzten Endgerät speichern. Zu anderen oder weitergehenden Nutzungen und Verwertungen sind Sie nicht berechtigt.

So ist es insbesondere unzulässig, eine elektronische oder gedruckte Kopie an Dritte weiterzugeben. Unzulässig und nicht erlaubt ist des Weiteren, das E-Book im Internet, in Intranets oder auf andere Weise zu verbreiten oder Dritten zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und jegliche den persönlichen Gebrauch übersteigende Vervielfältigung des E-Books ist ausdrücklich untersagt. Das vorstehend Gesagte gilt nicht nur für das E-Book insgesamt, sondern auch für seine Teile (z. B. Grafiken, Fotos, Tabellen, Textabschnitte).

Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte dürfen aus dem E-Book nicht entfernt werden, auch nicht das digitale Wasserzeichen.

## **Markenschutz**

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

## **Haftungsausschluss**

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

## Über den Autor

**Dr. Michael Kofler** studierte Telematik an der TU Graz. Er zählt zu den erfolgreichsten und vielseitigsten Computerbuchautoren im deutschen Sprachraum. Zu seinen Themengebieten zählen neben Linux auch Java, OS X, der Raspberry Pi, MySQL, KVM, Visual Basic und Excel-VBA. Viele seiner Bücher wurden übersetzt.

Michael Kofler arbeitet auch als Software-Entwickler, Berater sowie als Lehrbeauftragter an zwei Fachhochschulen.

<http://kofler.info/>